Министерство образования Республики Беларусь

Учреждение образования

Белорусский государственный университет информатики и радиоэлектроники

Факультет непрерывного и дистанционного обучения

Кафедра программного обеспечения информационных технологий

Электронный учебно-методический комплекс

по дисциплине

**МЕТРОЛОГИЯ, СТАНДАРТИЗАЦИЯ   
И СЕРТИФИКАЦИЯ   
В ИНФОРМАционных технологиях**

Для студентов специальности

**Информатика и технологии программирования**

Минск 2011

# Общие сведения

## Сведения об ЭУМК

Электронный учебно-методический комплекс по дисциплине «Метрология, стандартизация и сертификация в информационных технологиях» предназначен для студентов дистанционной формы обучения по специальности «Информатика и технологии программирования» (ИиТП).

Электронный учебно-методический комплекс составлен на основе рабочей учебной программы по курсу «Метрология, стандартизация и сертификация в информационных технологиях», утверждённой деканом факультета непрерывного и дистанционного обучения «\_\_\_» \_\_\_\_\_\_\_\_\_ 2010, регистрационный № УД \_\_\_\_\_\_\_\_\_\_\_\_/Р и рабочего учебного плана специальности 1-40 01 01 «Программное обеспечение информационных технологий».

**Составитель:**

**Л.А. Глухова,** доцент кафедры программного обеспечения информационных технологий Учреждения образования «Белорусский государственный университет информатики и радиоэлектроники», кандидат технических наук.

Рассмотрен и рекомендован к изданию на заседании кафедры программного обеспечения информационных технологий, протокол № \_\_\_   
от \_\_.\_\_.2010.

Одобрен и рекомендован к изданию Методической комиссией факультета Компьютерных систем и сетей, протокол № \_\_ от \_\_.\_\_.2010.

## Методические рекомендации по изучению дисциплины

Программа дисциплины «Метрология, стандартизация и сертификация в информационных технологиях» (МСиСвИТ) для студентов дистанционной формы обучения составлена в соответствии с требованиями образовательного стандарта ОСРБ 1-40 01 01-2007 по специальности 1-40 01 01 «Программное обеспечение информационных технологий» (ПОИТ).

При составлении структуры материалов по дисциплине преследовались цели обучения студентов дистанционной формы обучения, во-первых, основам стандартизации жизненного цикла ПС и систем; во-вторых, основам стандартизации оценки качества и метрологии ПС; в-третьих, основам сертификации ПС.

В состав *обучающего комплекса по дисциплине* МСиСвИР для студентов дистанционной формы обучения входят следующие компоненты:

* методические рекомендации по изучению дисциплины;
* рабочая учебная программа дисциплины;
* теоретический раздел (курс лекций);
* перечень вопросов для самопроверки по теоретическому разделу дисциплины;
* перечень вопросов к зачету по дисциплине;
* практический раздел, включающий
* теоретические и практические сведения к выполнению индивидуальных практическихработ;
* контрольные работы;
* индивидуальные практические работы.

***Курс лекций содержит следующие основные темы.***

* Жизненный цикл программных средств и систем: основные понятия и определения.
* СТБ ИСО/МЭК 12207-2003 – базовый стандарт Беларуси в области жизненного цикла программных средств и систем: структура жизненного цикла ПС и систем; основные, вспомогательные и организационные процессы жизненного цикла; адаптация требований стандарта СТБ ИСО/МЭК 12207-2003 к условиям проекта; инструментальные средства автоматизации жизненного цикла программных средств и систем.
* Качество программных средств: основные понятия и определения.
* Стандартизация качества программного обеспечения в Республике Беларусь: классификация методов определения показателей качества ПС; иерархическая модель оценки качества ПС, методы оценки качества ПС по стандартам ГОСТ 28195–99 и СТБ ИСО/МЭК 9126–2003.
* Стандартизация качества ПС за рубежом: стандарты серий ISO/IEC 9126 и ISO/IEC 14598; модели внешнего и внутреннего качества ПС, модель качества в использовании; метод оценки качества программных средств по ISO/IEC 14598–1:1999.
* Метрология качества программных средств: свойства и критерии обоснованности метрик качества ПС; внутренние и внешние метрики качества ПС; метрики качества ПС в использовании.
* Сертификация программных средств: основные понятия и определения в области технического нормирования, стандартизации и оценки соответствия; общие сведения об оценке соответствия в Республике Беларусь; организация сертификации ПС.

Для успешного освоения дисциплины «Метрология, стандартизация и сертификация в информационных технологиях» рекомендуется следующая ***последовательность ее изучения.***

1. ***Изучение теоретической темы.***

Темы желательно изучать в том порядке, в котором они представлены в лекционном курсе. Каждая последующая тема основана на знании материала предыдущих тем.

1. ***Выполнение контрольной работы по теме*** (если она предусмотрена).

По дисциплине предусмотрено две контрольных работы.

Вопросы п*ервой работы* связаны с организацией жизненного цикла ПС и оценкой качества ПС. Поэтому к выполнению контрольной работы № 1 нужно приступать после изучения следующих тем теоретического раздела: тема 1 (Жизненный цикл программных средств и систем. Основные понятия и определения), 2 (СТБ ИСО/МЭК 12207-2003 – базовый стандарт Беларуси в области жизненного цикла программных средств и систем), 3 (Качество программных средств. Основные понятия и определения), 4 (Стандартизация качества программного обеспечения в Республике Беларусь), 5 (Стандартизация качества программных средств за рубежом), 7 (Сертификация программных средств) Теоретического раздела.

*Вторая работа* связана с оценкой качества и метрологией ПС. К выполнению контрольной работы № 2 следует приступать после изучения следующих тем теоретического раздела: тема 5 (Стандартизация качества программных средств за рубежом), 6 (Метрология качества программных средств) Теоретического раздела.

Подробные пояснения к выполнению контрольных работ приведены в подразделе «Контрольные вопросы» Практического раздела.

***Правила оформления отчета по контрольной работе:***

Отчеты по контрольным работам выполняются в текстовом редакторе Word. Необходимо установить следующие размеры страницы:

Размер страницы – 21 х 29,7 см

Поле слева – 2,5 см

Поле справа – 1,5 см

Поле сверху – 2,5 см

Поле снизу – 2,2 см.

Размер шрифта Times New Roman 14, интервал между буквами обычный, интервал между строчками одинарный, выравнивание по ширине.

Объем каждой из контрольных работ должен составлять 4 - 5 страниц. Файл с контрольной работой № 1 должен иметь имя ФамилияК1.doc, с контрольной работой № 2 – ФамилияК2.doc.

1. ***Выполнение индивидуальной практической работы по теме*** (если она предусмотрена).

По дисциплине предусмотрены две индивидуальных практическихработы. Работы посвящены метрологии ПС. Перед выполнением работы необходимо изучить соответствующую тему Теоретического раздела. Выполнение работы заключается в написании программы, соответствующей полученному от преподавателя индивидуальному заданию, оценке ее метрик качества и написании отчета по работе.

***Правила оформления отчета по индивидуальной практической работе:***

Отчеты по работам выполняются в текстовом редакторе Word. Необходимо установить следующие размеры страницы:

Размер страницы – 21 х 29,7 см

Поле слева – 2,5 см

Поле справа – 1,5 см

Поле сверху – 2,5 см

Поле снизу – 2,2 см.

Размер шрифта Times New Roman 14, интервал между буквами обычный, интервал между строчками одинарный, выравнивание по ширине.

Объем каждой из индивидуальных практических работ должен составлять 5 - 7 страниц. Файл с индивидуальной практической работой № 1 должен иметь имя ФамилияИР1.doc, с индивидуальной практической работой № 2 – ФамилияИР2.doc.

Итоговой формой контроля знаний по дисциплине является ***зачет***. По каждому разделу и подразделу лекционного курса предусмотрены вопросы. Перечень вопросов к зачету приведен в ЭУМКД.

Студент допускается к зачету по дисциплине после успешной сдачи предусмотренных индивидуальных практическихи контрольных работ.

## Рабочая учебная программа

**Учреждение образования**

**«Белорусский государственный университет**

**информатики и радиоэлектроники»**

УТВЕРЖДАЮ

Декан факультета НиДО

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.М. Бондарик

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2011 г.

Регистрационный № УД-\_\_\_\_\_\_\_\_\_\_\_\_ /р.

МЕТРОЛОГИЯ, СТАНДАРТИЗАЦИЯ И СЕРТИФИКАЦИЯ   
В ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЯХ

Рабочая учебная программа для специальности:

Информатика и технологии программирования

Факультет непрерывного и дистанционного обучения

Кафедра Программное обеспечение информационных технологий

Курс 4

Индивидуальные практические работы – Зачет\_4 (курс)

2 (количество работ)

Контрольные работы\_2 \_\_(количество работ)

Всего часов Форма получения

по дисциплине\_\_98\_\_ высшего образования дистанционная

2011

Составила Глухова Лилия Александровна, к.т.н., доцент

Рабочая учебная программа составлена на основе учебной программы «Метрология, стандартизация и сертификация в информатике и радиоэлектронике», утвержденной ректором БГУИР 03.06.2008 г., регистрационный № УД-40-021/уч. и учебного плана специальности 1-40 01 01 «Программное обеспечение информационных технологий» (дистанционная форма получения высшего образования для студентов набора, начиная с 2009 года).

Рассмотрена и рекомендована к утверждению на заседании кафедры программного обеспечения информационных технологий

протокол № \_\_\_\_ от «\_\_\_\_» \_\_\_\_\_\_\_\_\_ 20\_\_\_ г.

Заведующий кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (В.В. Бахтизин)

Одобрена и рекомендована к утверждению Методической комиссией факультета Компьютерные системы и сети Учреждения образования «Белорусский государственный университет информатики и радиоэлектроники»

протокол № \_\_\_\_ от «\_\_\_\_» \_\_\_\_\_\_\_\_\_ 20\_\_\_ г.

Председатель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (М.И. Лукашевич)

СОГЛАСОВАНО

Начальник ОМОУП Ц.С.Шикова

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**Цель преподавания дисциплины.** Целью дисциплины является подготовка специалиста, владеющего фундаментальными знаниями и практическими навыками в области метрологии, стандартизации и сертификации программного обеспечения.

**Задачи изучения дисциплины.**

Задачами изучаемой дисциплины являются:

* изучение студентами требований и рекомендаций современных международных и национальных стандартов в области разработки программных средств;
* изучение студентами требований и рекомендаций современных международных и национальных стандартов в области оценки качества программных средств;
* ознакомление студентов с основами сертификации программных средств;
* приобретение студентами теоретических знаний в области метрологии и оценки качества программных средств.
* приобретение студентами практических навыков в области метрологии программных средств.

В результате изучения дисциплины студенты должны:

***знать:***

* основные национальные и международные стандарты в области жизненного цикла программных средств;
* основные национальные и международные стандарты в области оценки качества и метрологии программных средств;
* основы оценки соответствия в Республике Беларусь;
* основы организации сертификации программных средств;

***уметь:***

* разрабатывать модели качества для конкретных программных средств;
* выбирать и оценивать метрики качества для конкретных программных средств;

***приобрести навыки:***

* использования методов оценки качества программных средств;
* измерения метрик качества программных средств.

**Перечень дисциплин, усвоение которых необходимо для изучения данной дисциплины.**

|  |  |  |
| --- | --- | --- |
| № пп | Название дисциплины | Раздел, тема |
| 1 | Основы алгоритмизации и программирования | Все темы |
| 2 | Языки программирования | Все темы |

**СОДЕРЖАНИЕ ДИСЦИПЛИНЫ**

| № п/п | Название и содержание тем | Контроль- ная работа (номер и  тема по п.2) | Индивиду- альная  практическая  работа  (по п.1) | Оснащение контрольных и индивидуальных практических работ   (по п.4) | Литература  (по п.3) | Рекомендуемый объем  для изучения  (в часах)2 | Форма контроля знаний (зачет по контрольной работе, защита лабораторной работы, зачет) |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Раздел 1. Стандартизация жизненного цикла программного обеспечения** | | | | | | | |
| 1. | **Тема 1. Введение в стандартизацию программных средств.** Общие сведения о стандартах и об организациях, занимающихся стандартизацией программных средств и систем. Основные понятия и определения в области жизненного цикла (ЖЦ) программных средств (ПС) и систем. Обзор базовых международных и национальных стандартов в области жизненного цикла ПС. | 1 | — | 1 | Л1, разд. 1, 2, Л2, разд. 1, 2, Л5 | 6 | Зачет по контрольной работе |
| 2. | **Тема 2. Стандарт СТБ ИСО/МЭК 12207-2003 –** б**азовый стандарт в области жизненного цикла программного обеспечения.** Классификация процессов ЖЦ ПС. Процессы, работы и задачи ЖЦ ПС и систем. | 1 | — | 1 | Л1, разд. 2, Л2, разд. 2, Л5 | 18 | Зачет по контрольной работе |
| **Раздел 2. Стандартизация оценки качества программного обеспечения** | | | | | | | |
| 3. | **Тема 3. Стандартизация оценки качества программного обеспечения в Республике Беларусь.** Основные понятия и определения в области оценки качества программных средств. Обзор стандартов в области оценки качества ПС, действующих на территории РБ. ГОСТ 28195-99. СТБ ИСО/МЭК 9126-2003. | 1 | — | 1 | Л1, разд. 8, 9, Л2, разд. 3, 4, Л3, Л4, Л6 | 19 | Зачет по контрольной работе |
| 4. | **Тема 4. Стандартизация оценки качества программного обеспечения за рубежом.** Серия стандартов ISO/IEC 9126-1-4:2001-2004. Серия стандартов ISO/IEC 14598-1-6:1998-2001. | 2 | — | 1 | Л1, разд. 10, Л2, разд. 5, Л7 – Л11, Л31 | 20 | Зачет по контрольной работе |
| **Раздел 3. Метрология программного обеспечения** | | | | | | | |
| 5. | **Тема 5. Метрология качества программных средств.** Метрики внутреннего, внешнего и качества в использовании ПС. Свойства метрик | 2 | 1, 2 | 1 – 6 | Л1, разд. 10, Л2, разд. 6, Л8 – Л11 | 22 | Зачеты по контрольной и лабораторным работам |
| **Раздел 4. Сертификация программного обеспечения** | | | | | | | |
| 6. | **Тема 6. Основы сертификации программных средств.** Основные понятия и определения в области технического нормирования, стандартизации и оценки соответствия. Основополагающие документы Национальной системы подтверждения соответствия Республики Беларусь. Основы сертификации программных средств. | 1 | — | 1 | Л1, разд. 11, Л2, разд. 7, Л12 – Л13 | 13 | Зачет по контрольной работе |
| **Зачет** | | | | | | | |

**1. ИНДИВИДУАЛЬНЫЕ ПРАКТИЧЕСКИЕ РАБОТЫ,   
ИХ ХАРАКТЕРИСТИКА**

|  |  |  |  |
| --- | --- | --- | --- |
| № пп | Тема | Содержание | Объем в часах |
| 1. | Метрики сложности потока управления программ | Разработка схемы алгоритма и оценка ее сложности на основе метрик потока управления программ. Индивидуальная практическая работа должна быть оформлена в формате Word в соответствии с общими требованиями к оформлению.  Вариант индивидуальной практической работы рассчитывается студентом по номеру зачетной книжки.  Объем работы ≈ 5 – 7 стр. | 4 |
| 2. | Метрики размера программ. Метрики потока данных программ | Разработка программы и оценка ее сложности на основе метрик размера программ и потока данных программ. Индивидуальная практическая работа должна быть оформлена в формате Word в соответствии с общими требованиями к оформлению.  Вариант индивидуальной практической работы рассчитывается студентом по номеру зачетной книжки.  Объем работы ≈ 5 – 7 стр. | 4 |

**2. КОНТРОЛЬНЫЕ РАБОТЫ, ИХ ХАРАКТЕРИСТИКА**

| № пп | Тема | Характеристика | Рекомендуемый объем в часах |
| --- | --- | --- | --- |
| 1. | Жизненный цикл программных средств и оценка их качества | Задание по контрольной работе представляет собой перечень вопросов, связанных с жизненным циклом ПС и оценкой их качества. Необходимо ответить на вопросы в достаточно полном объеме. Контрольная работа должна быть оформлена в формате Word в соответствии с общими требованиями к оформлению.  Вариант контрольной работы рассчитывается студентом по номеру зачетной книжки.  Объем работы ≈ 4 – 5 стр. | 4 |
| 2. | Оценка качества и метрология программных средств | Задание по контрольной работе заключается в разработке метрики качества, соответствующей индивидуальному заданию, и определении критериев обоснованности и свойств разработанной метрики. Контрольная работа должна быть оформлена в формате Word в соответствии с общими требованиями к оформлению.  Вариант контрольной работы рассчитывается студентом по номеру зачетной книжки.  Объем работы ≈ 4 – 5 стр. | 4 |

**3. ЛИТЕРАТУРА**

**3.1. ОСНОВНАЯ**

1. Бахтизин, В. В. Стандартизация и сертификация программного обеспечения : учеб. пособие / В. В. Бахтизин, Л. А. Глухова. – Минск : БГУИР, 2006.
2. ЭУМКД «Метрология, стандартизация и сертификация в информатике и радиоэлектронике» для дистанционной формы обучения специальности I-40 01 01 Программное обеспечение информационных технологий. – М.: БГУИР, 2011.
3. ГОСТ 28195–99. Оценка качества программных средств. Общие положения.
4. ГОСТ 28806–90. Качество программных средств. Термины и определения.
5. СТБ ИСО/МЭК 12207–2003. Информационная технология. Процессы жизненного цикла программных средств.
6. СТБ ИСО/МЭК 9126–2003. Информационные технологии. Оценка программной продукции. Характеристики качества и руководства по их применению.
7. ISO/IEC 14598***–***1:1999. Информационная технология – Оценка программного продукта – Часть 1: Общий обзор.
8. ISO/IEC 9126–1:2001. Программная инженерия – Качество продукта – Часть 1: Модель качества.
9. ISO/IEC TR 9126–2:2003. Программная инженерия – Качество продукта – Часть 2: Внешние метрики.
10. ISO/IEC TR 9126–3:2003. Программная инженерия – Качество продукта – Часть 3: Внутренние метрики.
11. ISO/IEC TR 9126–4:2004. Программная инженерия – Качество продукта – Часть 4: Метрики качества в использовании.
12. Закон Республики Беларусь от 5 января 2004 г. № 262-З «О техническом нормировании и стандартизации».
13. Закон Республики Беларусь от 5 января 2004 г. № 269-З «Об оценке соответствия требованиям технических нормативных правовых актов в области технического нормирования и стандартизации».

**3.2. Дополнительная**

1. ГОСТ 19.701-90 – Единая система программной документации – Схемы алгоритмов, программ, данных и систем – Условные обозначения и правила выполнения.
2. ГОСТ Р ИСО/МЭК 15910–2002. Информационная технология. Процесс создания документации пользователя программного средства.
3. ГОСТ Р ИСО/МЭК ТО 15271–2002. Информационная технология. Руководство по применению ГОСТ Р ИСО/МЭК 12207 (Процессы жизненного цикла программных средств).
4. ГОСТ Р ИСО/МЭК ТО 16326–2002. Программная инженерия. Руководство по применению ГОСТ Р ИСО/МЭК 12207 при управлении проектом.
5. СТБ ИСО/МЭК 14764–2003. Информационная технология. Сопровождение программных средств.
6. ISO 10006–1997. Управление качеством – Руководства по качеству при управлении проектом.
7. ISO/IEC 12207:1995. Информационная технология – Процессы жизненного цикла программных средств.
8. ISO/IEC 12207:2008. Системная и программная инженерия – Процессы жизненного цикла программных средств. – Введ. 2008-02-01. – Нью-Йорк : ISO/IEC-IEEE, 2008.
9. ISO/IEC 14598–2:2000. Программная инженерия – Оценка продукта – Часть 2: Планирование и управление.
10. ISO/IEC 14598–3:2000. Программная инженерия – Оценка продукта – Часть 3: Процесс для разработчиков.
11. ISO/IEC 14598–4:1999. Программная инженерия – Оценка продукта – Часть 4: Процесс для заказчиков.
12. ISO/IEC 14598–5:1998. Информационная технология – Оценка программного продукта – Часть 5: Процесс для оценщиков.
13. ISO/IEC 14598–6:2001. Программная инженерия – Оценка продукта – Часть 6: Документация модулей оценки.
14. ISO/IEC 15846:1998. Информационная технология – Процессы жизненного цикла программных средств – Управление конфигурацией.
15. ISO/IEC 15939:2002. Программная инженерия – Процесс измерения программных средств.
16. ISO/IEC 16085:2004. Информационная технология – Процессы жизненного цикла программных средств – Управление рисками.
17. Изосимов А.В., Рыжко А.Л. Метрическая оценка качества программ. – М.: МАИ, 1989.
18. Липаев В.В. Обеспечение качества программных средств. Методы и стандарты. – М.: СИНТЕГ, 2001.
19. [www.gosstandart.gow.by](http://www.gosstandart.gow.by/)

**4. ПЕРЕЧЕНЬ КОМПЬЮТЕРНЫХ ПРОГРАММ, НАГЛЯДНЫХ И ДРУГИХ ПОСОБИЙ, МЕТОДИЧЕСКИХ УКАЗАНИЙ И МАТЕРИАЛОВ И ТЕХНИЧЕСКИХ СРЕДСТВ ОБУЧЕНИЯ**

**Перечень методических материалов и пособий**

1. ЭУМКД «Метрология, стандартизация и сертификация в информатике и радиоэлектронике» для дистанционной формы обучения специальности I-40 01 01 Программное обеспечение информационных технологий. – М.: БГУИР, 2011.
2. Глухова Л.А., Фадеева Е.П., Фадеева Е.Е., Болтак С.В. Основы алгоритмизации и программирования: Лаб. практикум для студ. спец. I-40 01 01 «Программное обеспечение информационных технологий» дневной формы обуч. В 4 ч. Ч.1. – Мн.: БГУИР, 2004.
3. Глухова Л.А., Фадеева Е.П., Фадеева Е.Е. Основы алгоритмизации и программирования: Лаб. практикум для студ. спец. I-40 01 01 «Программное обеспечение информационных технологий» дневной формы обуч. В 4 ч. Ч.2. – Мн.: БГУИР, 2005.
4. Глухова Л.А., Фадеева Е.П., Фадеева Е.Е. Основы алгоритмизации и программирования: Лаб. практикум для студ. спец. I-40 01 01 «Программное обеспечение информационных технологий» дневной формы обуч. В 4 ч. Ч.3. – Мн.: БГУИР, 2007.

**Перечень компьютерных программ, необходимых для выполнения индивидуальных практических работ**

5. Операционная система Windows.

6. Система программирования Borland Pascal 7.0.

**Протокол согласования учЕбной программы**

**по изучаемой учебной дисциплине**

**С ДРУГИМИ ДИСЦИПЛИНАМИ СПЕЦИАЛЬНОСТИ**

(дисциплины, изучение которых опирается на данную дисциплину)

|  |  |  |  |
| --- | --- | --- | --- |
| Название дисциплины,  с которой требуется согласование | Кафедра, обеспечивающая изучение этой дисциплины | Предложения об изменениях в содержании учебной программы по изучаемой дисциплине | Решение, принятое кафедрой, разработавшей учебную программу (с указанием даты и номера протокола)1 |
| 1 | 2 | 3 | 4 |
| Технологии разработки программного обеспечения | ПОИТ | отсутствуют |  |

Зав. кафедрой ПОИТ (В.В.Бахтизин)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1 При наличии предложений об изменениях в содержании учебной программы по изучаемой учебной дисциплине.

# Теоретический раздел

## Введение

Широкое внедрение информационных технологий во все сферы деятельности человека приводит к разработке огромного количества программных средств (ПС) различного функционального назначения. Это вызывает необходимость в создании международных и национальных стандартов в области программных средств, систем и связанных с ними процессов. Применение стандартов позволяет унифицировать процессы разработки, эксплуатации и сопровождения ПС, что способствует повышению качества процессов и самих ПС и повышению конкурентоспособности ПС на внутреннем и внешнем рынках.

Основной международной организацией, занимающейся деятельностью по стандартизации, является *Международная организация по стандартизации ИСО* (International Standards Organization, ISO). Стандарты, принимаемые данной организацией, получают аббревиатуру ISO и имеют ранг международных.

Области электротехники, электроники, радиосвязи, приборостроения не входят в компетенцию ISO. Стандартизацией в данных областях занимается *Международная электротехническая комиссия МЭК* (International Electrotechnical Commission, IEC). В области стандартизации информационных технологий ISO и IEC объединили свою деятельность, создав *Объединенный технический комитет 1* (Joint Technical Committee 1, JTC1).

В Республике Беларусь республиканским органом государственного управления по вопросам технического нормирования, стандартизации, метрологии и оценки соответствия является Комитет по стандартизации, метрологии и сертификации при Совете Министров Республики Беларусь (*Госстандарт*).

В систему Госстандарта входит *Белорусский государственный институт стандартизации и сертификации* (*БелГИСС*), занимающийся вопросами технического нормирования, стандартизации, подтверждения соответствия, управления качеством, нормативно-технического и информационного обеспечения.

За рубежом и в Беларуси приняты следующие ***обозначения стандартов***:

* вначале записывается категория стандарта; например, СТБ – стандарт Беларуси, ГОСТ Р – государственный стандарт России, ГОСТ – межгосударственный стандарт для ряда стран СНГ, до распада СССР аббревиатура ГОСТ обозначала государственный стандарт СССР, ISO/IEC (ИСО/МЭК) – международный стандарт организаций ISO и IEC;
* если стандарт разработан методом прямого применения (например, является аутентичным переводом международного стандарта), то за категорией стандарта следует обозначение категории базового стандарта (в русском именовании); например, СТБ ИСО/МЭК и ГОСТ Р ИСО/МЭК – это аутентичные переводы международного стандарта ISO/IEC, СТБ ГОСТ Р – стандарт Республики Беларусь, разработанный методом прямого применения стандарта России;
* затем следует номер стандарта; при этом, если стандарт разработан методом прямого применения, то его номер совпадает с номером базового стандарта; например, базовый стандарт ISO/IEC 9126, аутентичные стандарты Беларуси СТБ ИСО/МЭК 9126 и России ГОСТ РИСО/МЭК 9126;
* если стандарт состоит из нескольких частей, то после номера стандарта записывается номер его части; например, ISO/IEC 14598–1;
* после номера стандарта (или его части) записывается год его утверждения; например, ISO/IEC 9126:1991, СТБ ИСО/МЭК 9126–2003;
* если стандарт еще находится в стадии разработки, но имеется необходимость в опубликовании его материалов, то в обозначении стандартов ISO/IEC, работа над которыми начата до 1999 г., после категории записывается аббревиатура TR (Technical Report – Технический отчет, ТО); например, ISO/IEC TR 15271:1998, ГОСТ РИСО/МЭК ТО 15271–2002.

Существующие стандарты можно разделить на ***два основных типа***: *стандарты на продукты*, определяющие их характеристики и требования к ним; *стандарты на процессы*, определяющие конкретные методы создания продуктов. В пособии рассматриваются стандарты обоих типов.

Теоретический раздел состоит из семи разделов.

**Разделы 1 – 2** посвящены изучению современных стандартов в области жизненного цикла (ЖЦ) ПС и систем, действующих за рубежом и в Беларуси. Стандартизация процессов ЖЦ ПС и систем занимает важное место в стандартизации информационных технологий и программной инженерии. Строгое соблюдение стандартов, связанных с ЖЦ ПС, обеспечивает улучшение технико-экономических показателей проектов ПС, позволяет унифицировать процесс и технологии разработки ПС. Это приводит к существенному повышению качества как процессов ЖЦ ПС, так и самих программных продуктов (ПП).

В **первом разделе** рассматриваются основные понятия и определения в области ЖЦ ПС и систем.

Во **втором разделе**детально рассмотрены и пояснены требования и рекомендации базового стандарта Беларуси в области ЖЦ ПС и систем ***СТБ ИСО/МЭК 12207–2003***, являющегося аутентичным аналогом международного стандарта ***ISO/IEC 12207:1995***. Рассмотрены процессы, работы и задачи ЖЦ ПС и систем. Пояснены возможности адаптации требований стандарта *СТБ ИСО/МЭК 12207–2003*к условиям конкретного проекта.

**Разделы 3 – 5** посвящены изучению современных стандартов в области оценки качества ПС. Применение компьютеров становится все разнообразнее. Их корректная работа, определяемая в первую очередь программным обеспечением, часто является критичной для здоровья и безопасности человека, успеха предприятий и организаций. В этой связи первостепенное значение имеет разработка ПП высокого качества, ключевыми факторами которого являются тщательное специфицирование и оценка качества промежуточных и конечного ПП.

В **третьем разделе** рассматриваются основные понятия и определения в области качества программных средств.

**Четвертый раздел** посвящен изучению основных положений стандартов в области оценки качества ПС, действующих на территории Республики Беларусь. Рассмотрены требования и рекомендации стандартов ***ГОСТ 28806–90***, ***ГОСТ 28195–99***, ***СТБ ИСО/МЭК 9126–2003***. Рассмотрены модели и методы оценки качества ПС, определенные в данных стандартах.

В **пятом разделе** рассматриваются современные международные стандарты в области оценки качества ПС. Основными из них являются серии стандартов ***ISO/IEC 9126–1–4:2001–2004*** и ***ISO/IEC 14598–1–6:1998–2001***. В стандартах *ISO/IEC 9126****–****1****–****4:2001****–****2004* регламентируется иерархическая модель качества ПС, состоящая из уровней характеристик, подхарактеристик и метрик. В стандартах *ISO/IEC 14598****–****1****–****6:1999****–****2001* определен процесс оценки качества ПС и его модификации для различных целей оценки.

**Раздел 6** посвящен вопросам метрологии программного обеспечения. Рассматриваются современные подходы, используемые в метрологии качества программных средств. Рассмотрены желательные свойства и критерии обоснованности метрик качества ПС, определенные в серии стандартов *ISO/IEC 9126–2–4:2003–2004.* Приведены примеры внутренних, внешних метрик качества ПС, а также метрик качества в использовании.

**Раздел 7** посвящен вопросам сертификации ПС. Сертификация ПС предназначена для обеспечения защиты жизни и здоровья человека, защиты имущества, охраны окружающей среды при использовании ПС, для повышения конкурентоспособности ПС и создания благоприятных условий для обеспечения свободного перемещения ПС на внутреннем и внешнем рынках. Рассмотрено положение дел в Республике Беларусь в области оценки соответствия продукции требованиям технических нормативных правовых актов. Описаны основные законы Республики Беларусь, посвященные техническому нормированию, стандартизации и оценке соответствия. Рассмотрены формы подтверждения соответствия (обязательная и добровольная сертификация и декларирование соответствия). Описана организация сертификации ПС.

***В теоретическом разделе используются следующие сокращения:***

ВТО – Всемирная торговая организация;

ЖЦ – жизненный цикл;

НСПС РБ – Национальная система подтверждения соответствия Республики Беларусь;

ПП – программный продукт;

ПС – программное средство.

## 1. Жизненный цикл программных средств и систем. Основные понятия и определения

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.

В настоящее время во всем мире ведутся активные работы в направлении стандартизации ЖЦ ПС и систем. Стандартизация ЖЦ позволяет упорядочить вопросы создания, сопровождения и управления ПС. Строгое соблюдение требований стандартов обеспечивает унификацию процессов ЖЦ программных средств и их компонентов. Это ведет к повышению качества процессов ЖЦ и в конечном итоге к повышению качества самих программных средств.

С понятием ЖЦ ПС тесно связаны следующие ***основные термины***, соответствующие определениям стандартов *ISO/IEC 2382–1:1993, ISO/IEC 2382–20:1990, ISO 8402:1994, СТБ ИСО/МЭК 12207–2003*.

*Аттестация (validation):* подтверждение экспертизой и представлением объективных доказательств того, что конкретные требования к конкретным объектам полностью реализованы. В процессе проектирования и разработки аттестация связана с экспертизой продукта в целях определения его соответствия потребностям (требованиям) пользователя.

*Аудит (audit):* проверка, выполняемая компетентным органом (лицом) с целью обеспечения независимой оценки степени соответствия программных продуктов или процессов установленным требованиям.

*Базовая линия (baseline):* официально принятая версия элемента конфигурации, независимая от среды, формально обозначенная и зафиксированная в конкретный момент времени жизненного цикла элемента конфигурации.

*Верификация (verification):* подтверждение экспертизой и представлением объективных доказательств того, что конкретные требования полностью реализованы. В процессе проектирования и разработки верификация связана с экспертизой результатов данной работы для определения их соответствия установленным к данной работе требованиям.

*Квалификационное испытание (qualification testing):* испытание (тестирование), проводимое разработчиком, при необходимости санкционированное заказчиком, для демонстрации того, что программный продукт удовлетворяет установленным требованиям и готов к использованию в заданных условиях эксплуатации.

*Квалификационное требование (qualification requirement):* набор критериев или условий, которые должны быть удовлетворены для того, чтобы квалифицировать программный продукт на соответствие установленным требованиям и готовность к использованию в заданных условиях эксплуатации.

*Контекст использования (context of use):*пользователи, задания, среда (аппаратное обеспечение, программное обеспечение и материалы), а также физические и социальные среды, в которых используется данное прораммное средство.

*Обеспечение качества (quality assurance):* все запланированные и систематически выполняемые в рамках системы качества работы; при необходимости объективные доказательства, обеспечивающие уверенность в том, что объект будет полностью соответствовать установленным требованиям качества. Существуют как внешние, так и внутренние цели обеспечения качества. Внутреннее обеспечение качества создает уверенность у руководства в достижении заданных требований качества внутри организации. Внешнее обеспечение качества создает уверенность у потребителя или других лиц в достижении заданных требований качества в договорных или других ситуациях.

*Программная услуга (software servise):* выполнение работ, заданий или обязанностей, связанных с программным продуктом, таких, как разработка, сопровождение или эксплуатация.

*Программное обеспечение (программное средство, software):* полный набор (программное обеспечение) или часть (программное средство) программ, процедур, правил и связанной с ними документации системы обработки информации.

*Программный модуль (software unit):* отдельно компилируемая часть программного кода (программы).

*Программный продукт (software product):* набор компьютерных программ, процедур, связанных с ними документации и данных. Продукты включают промежуточные продукты и продукты, предназначенные для пользователей типа разработчиков и персонала сопровождения.

*Система (system):* комплекс, состоящий из процессов, технических и программных средств, устройств и персонала, обладающий возможностью удовлетворять установленным потребностям или целям.

*Система управления качеством (система менеджмента качества, система качества, quality management system, quality system):* часть общей системы управления, включающая организационную структуру, планирование, ответственность, методы, процедуры, процессы, ресурсы, необходимые для обеспечения качества продукции и (или) услуг. Как правило, система управления качеством является частью системы управления предприятием или организацией.

*Спецификация требований к программному продукту* *(software product specification):* определение и перечень требований к программному продукту.

*Тестируемость (testability):* степень, до которой могут быть запланированы объективность и реализуемость тестирования, проверяющего соответствие требованию.

*Тестовое покрытие (test coverage):* степень, до которой с помощью контрольных примеров проверяют требования к системе или программному продукту.

*Техническое задание (statement of work):* документ, используемый заказчиком в качестве средства для описания и определения задач, выполняемых при реализации договора.

*Элемент конфигурации (configuration item):* объект внутри конфигурации, который удовлетворяет функции конечного использования и может быть однозначно определен в данной эталонной точке.

## 2. СТБ ИСО/МЭК 12207-2003 – базовый стандарт Беларуси в области жизненного цикла программных средств и систем

### 2.1. Общие сведения о стандартах в области жизненного цикла программных средств и систем

В настоящее время базовым стандартом в области жизненного цикла программных средств и систем является международный стандарт ***ISO/IEC 12207: 2008 – Системная и программная инженерия – Процессы жизненного цикла программных средств***.

В Республике Беларусь c 2004 г. действует национальный стандарт ***СТБ ИСО/МЭК 12207-2003* – *Информационная технология – Процессы жизненного цикла программных средств***. Данный стандарт является аутентичным аналогом предыдущей редакции международного стандарта ***ISO/IEC 12207: 1995***.

В России в 2000 г. введен в действие ***ГОСТ Р ИСО/МЭК 12207–99***, содержащий аутентичный текст международного стандарта *ISO/IEC 12207: 1995*.

Стандарт *СТБ ИСО/МЭК 12207-2003*определяет жизненный цикл ПС и систем в целом. При этом под ***системой*** подразумевается комплекс, состоящий из процессов, технических и программных средств и персонала, обладающий возможностью удовлетворять установленным потребностям или целям. Однако стандарт *СТБ ИСО/МЭК 12207-2003* 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 (). 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.

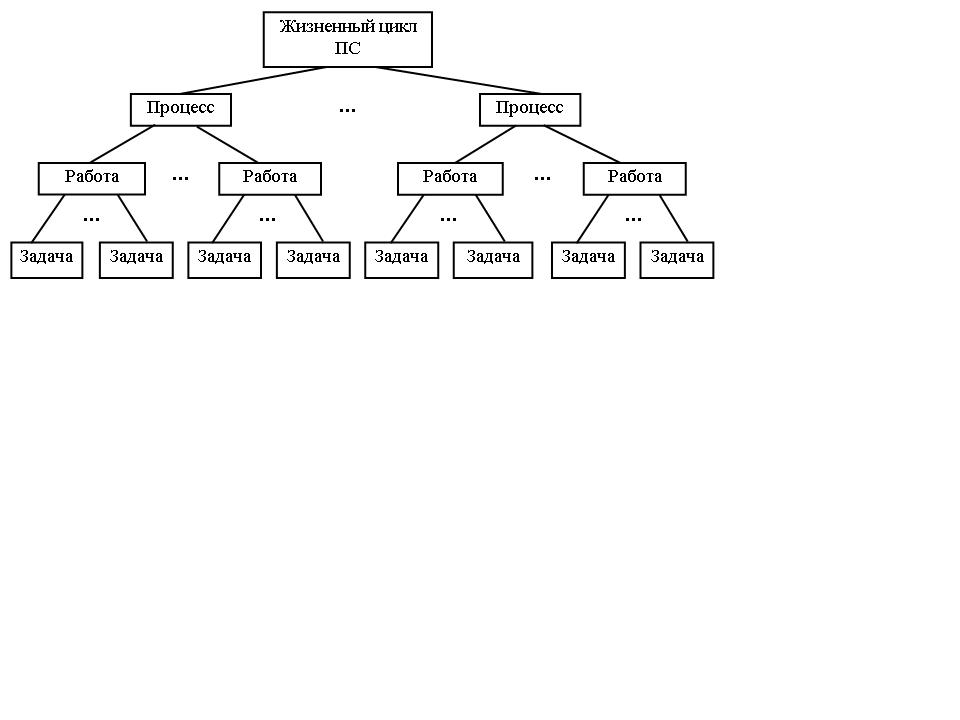


Рис. . Общая структура жизненного цикла программных средств

Процессы ЖЦ ПС делятся на ***три группы*** (рис. 2):

* основные;
* вспомогательные;
* организационные.

Нумерация групп процессов на приведена в соответствии со структурой стандарта *СТБ ИСО/МЭК 12207-2003*.

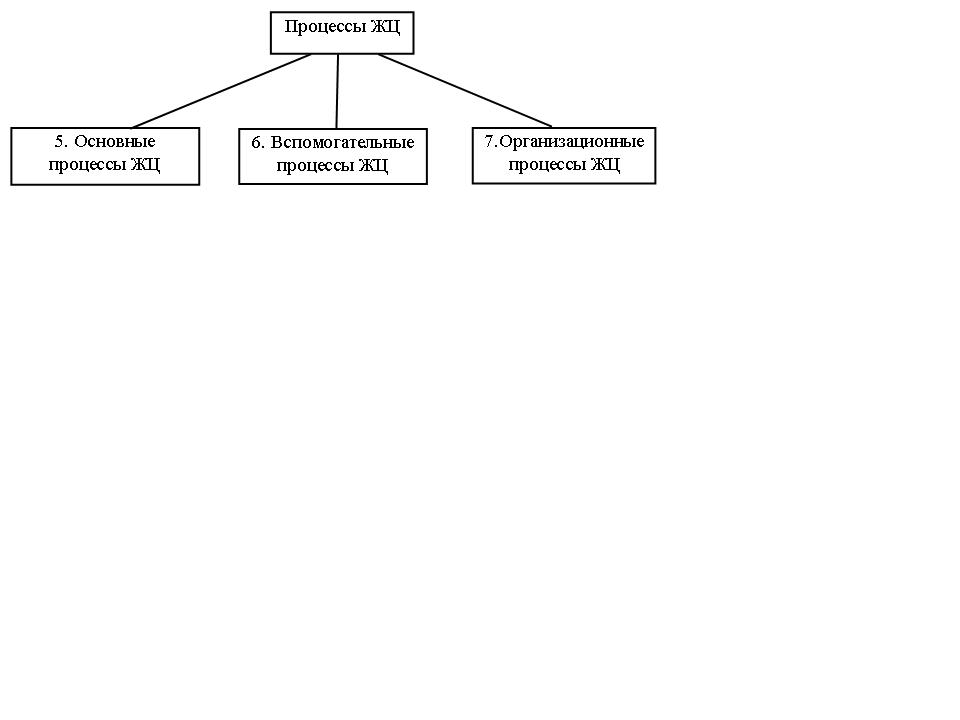


Рис. . Группы процессов жизненного цикла

На рис. 3 изображены процессы ЖЦ ПС с учетом их распределения по группам.

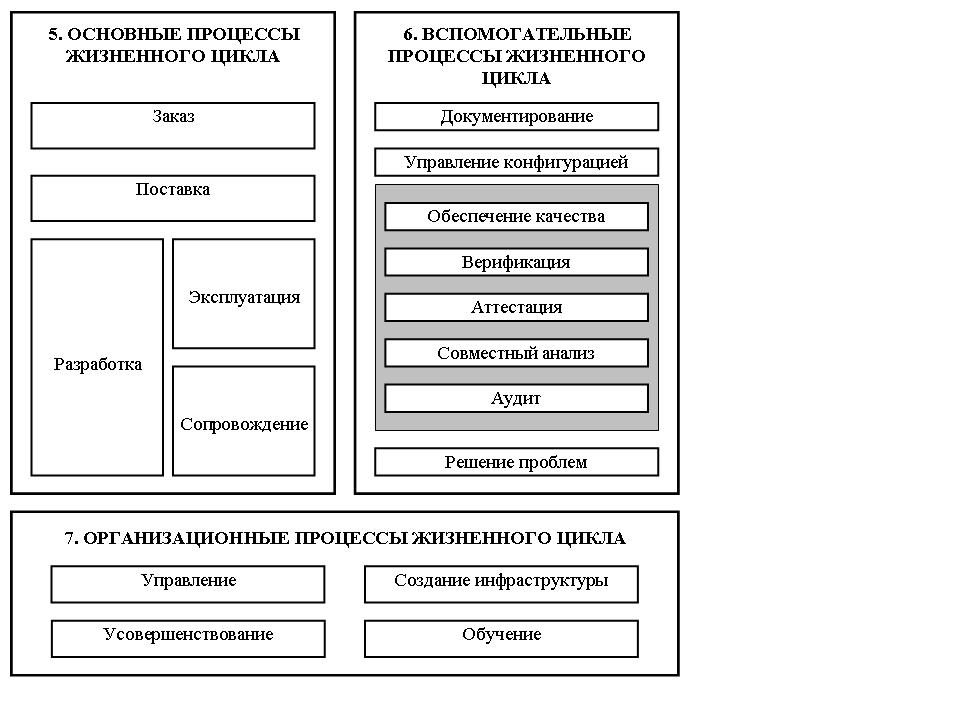


Рис. . Процессы жизненного цикла программных средств

**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.

***Заказчик*** –это организация, которая приобретает систему, программный продукт (ПП) или программную услугу.

***Поставщик*** – это организация, которая поставляет систему, ПП или программную услугу заказчику.

***Разработчик*** – это организация, которая разрабатывает ПП.

***Оператор*** – это организация, которая производит эксплуатационное обслуживание системы, содержащей ПП, в заданных условиях.

***Персонал сопровождения*** – это организация, которая предоставляет услуги по сопровождению программного продукта.

Как видно из рис. 3, основные процессысостоят из *пяти процессов*:

* заказ;
* поставка;
* разработка;
* эксплуатация;
* сопровождение.

**Вспомогательные процессы жизненного цикла**– это процессы, являющиеся целенаправленными составными частями других процессов. Их основное назначение – обеспечить успешную реализацию и качество выполнения программного проекта. Вспомогательный процесс инициируется и используется другим процессом.

Вспомогательные процессы состоят из *восьми процессов*:

* документирование;
* управление конфигурацией;
* обеспечение качества;
* верификация;
* аттестация;
* совместный анализ;
* аудит;
* решение проблем.

Для управления качеством программных средств в ходе жизненного цикла служат процессы обеспечения качества, верификации, аттестации, совместного анализа и аудита. Данные процессы на выделены серым прямоугольником. При этом процессы верификации, аттестации, совместного анализа и аудита могут реализовываться различными сторонами независимо или использоваться как методы процесса обеспечения качества.

Ответственность за работы и задачи вспомогательного процесса несет организация, выполняющая данный процесс.

**Организационные процессы жизненного цикла** – это процессы, предназначенные для создания в некоторой организации и совершенствования организационных структур, охватывающих процессы ЖЦ и соответствующий персонал. Обычно организационные процессы являются типовыми.

К организационным процессам относятся *четыре процесса*:

* управление;
* создание инфраструктуры;
* усовершенствование;
* обучение.

Ответственность за работы и задачи организационного процесса несет организация, выполняющая данный процесс.

Управление процессами жизненного цикла на проектном уровне осуществляется в соответствии с процессом управления. Управление этими процессами на организационном уровне выполняется в соответствии с процессами усовершенствования и обучения. Инфраструктура процессов ЖЦ определяется в соответствии с процессом создания инфраструктуры.

Адаптация данных процессов к условиям проекта осуществляется в соответствии с процессом адаптации (см. подразд. 2.5).

Действия по управлению, созданию инфраструктуры и адаптации процессов жизненного цикла выполняют те организации, которые реализуют соответствующие процессы ЖЦ.

Следует обратить внимание на то, что взаимоотношения между процессами, определяемые в *СТБ ИСО/МЭК 12207-2003*, всегда статические. В реальной же жизни отношения между процессами и участниками программного проекта являются динамическими. Каждый процесс и выполняющая его организация включаются в проект уникальным образом.

*Процесс заказа и заказчик* включаются в проект при определении системы, которая должна содержать программный продукт.

*Процесс поставки и поставщик* включаются при предоставлении программного продукта или услуги, от которых зависит система.

*Процесс разработки и разработчик* включаются в проект при анализе системы для корректного выделения и определения программного продукта, при его разработке и обеспечении подключения к системе.

*Процесс эксплуатации и оператор* включаются при эксплуатации программного продукта в системной среде в интересах пользователей.

*Процесс сопровождения и персонал сопровождения* включаются при сопровождении и поддержке программного продукта в эксплуатационной готовности и для консультаций пользователей.

*Вспомогательный или организационный процесс* включается при необходимости обеспечения уникальных специализированных функций для других процессов.

### 2.2. Основные процессы жизненного цикла

Основные процессы жизненного цикла программных средств и систем представлены на рис. 4. На данном рисунке сохранена нумерация процессов, принятая в *СТБ ИСО/МЭК 12207-2003*.

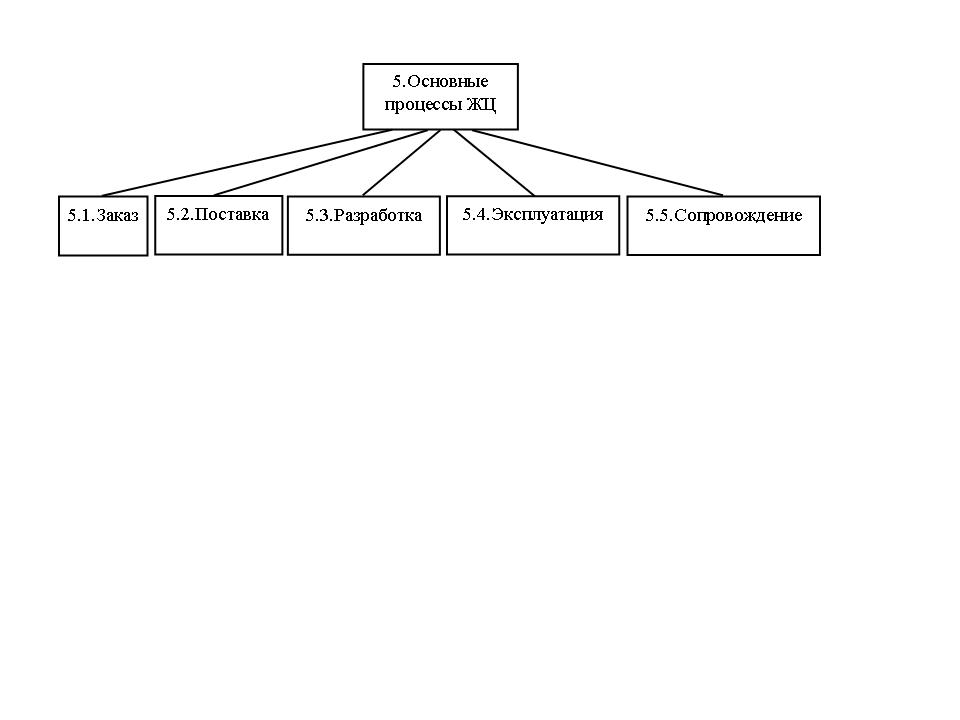


Рис. . Основные процессы жизненного цикла программных средств и систем

2.2.1. Процесс заказа (The Acquisition Process)

***Процесс заказа*** определяет работы и задачи заказчика. Процесс заказа состоит из определения потребностей заказчика в системе, программном продукте или программной услуге, подготовки и выпуска заявки на подряд, выбора поставщика и управления процессом заказа до завершения приемки системы, программного продукта или программной услуги.

Процесс заказа состоит из *пяти* *работ* (). Здесь и на последующих рисунках сохранена нумерация работ, принятая в *СТБ ИСО/МЭК 12207-2003.* Общее число задач по данным работам равно 23.

Табл. 1 содержит задачи, реализуемые при выполнении соответствующих работ процесса заказа, и типы выходных результатов данных задач, которые должны быть документально оформлены.

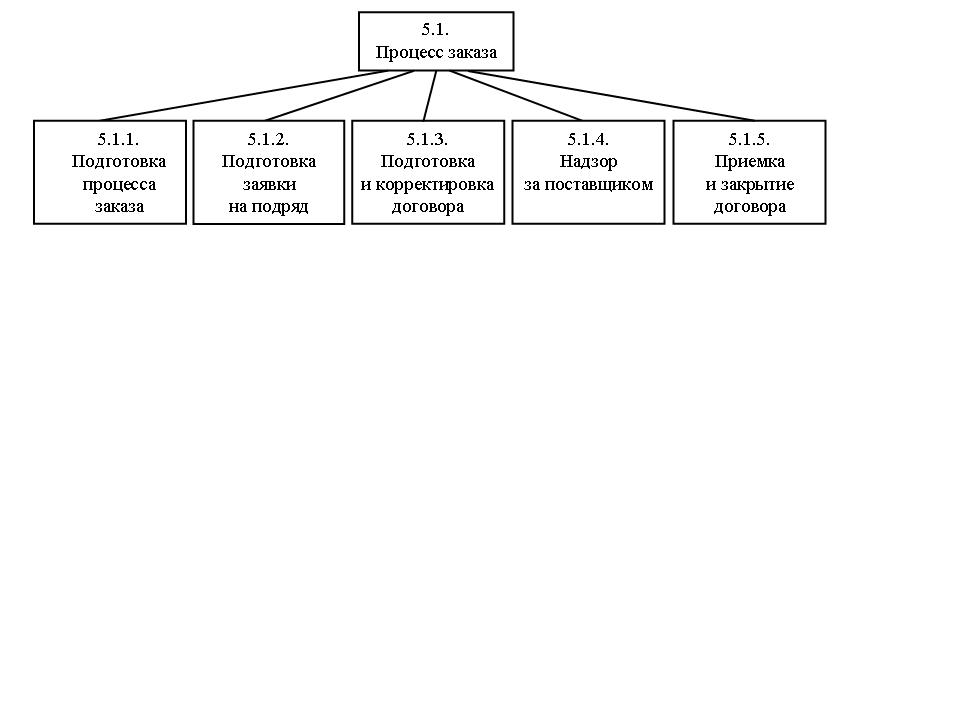


Рис. . Структура процесса заказа

Таблица 1

**Задачи работ процесса заказа**

|  |  |  |
| --- | --- | --- |
| № задачи в ИСО/МЭК 12207 | Описание задачи | Тип выходного  результата задачи |
| 5.1.1. Задачи работы ***Подготовка процесса заказа*** | | |
|  | Описание концепции или потребности в заказе | Описание  (в составе результата  задачи 5.1.1.8) |
|  | Анализ требований к системе | Описание  (в составе результата 5.1.1.8) |
|  | Согласование проанализированных требований (выполняется, если анализ требований осуществляет не заказчик, а поставщик) |  |
|  | Анализ требований к программным средствам (выполняется заказчиком или, по его поручению, поставщиком) | Описание  (в составе результата 5.1.1.8) |
|  | Использование процессов разработки при выполнении задач 5.1.1.2 и 5.1.1.4 |  |
|  | Рассмотрение вариантов реализации заказа (покупка готового ПП, разработка собственными силами, разработка на договорной основе, модернизация существующего ПП, комбинация вариантов) | Описание  (в составе результата 5.1.1.8) |
|  | Оценка условий приобретения готового программного продукта |  |
|  | Документальное оформление и выполнение плана заказа | План |
|  | Определение и документальное оформление правил и условий реализации договора | Описание |
| 5.1.2. Задачи работы ***Подготовка заявки на подряд*** | | |
|  | Документальное оформление требований к заказу (заявки на подряд) | Описание |
|  | Адаптация *СТБ ИСО/МЭК 12207-2003* к условиям проекта | Описание  (в составе результата 5.1.2.1) |
|  | Определение контрольных точек договора | Описание  (в составе результата 5.1.2.1) |
|  | Предоставление требований к заказу поставщикам для тендера (конкурса) |  |
| 5.1.3. Задачи работы ***Подготовка и корректировка договора*** | | |
|  | Определение процедуры выбора поставщика на тендерной (конкурсной) основе | Процедура |
|  | Выбор поставщика |  |
|  | Окончательное решение по адаптации *СТБ ИСО/МЭК 12207-2003* к условиям проекта | Описание  (в составе результата 5.1.3.4) |
|  | Подготовка и заключение договора с поставщиком | Договор |
|  | Контроль изменений, вносимых в договор, в ходе его реализации |  |
| 5.1.4. Задачи работы ***Надзор за поставщиком*** | | |
|  | Надзор за работами поставщика |  |
|  | Взаимодействие с поставщиком для решения проблем |  |
| 5.1.5. Задачи работы ***Приемка и закрытие договора*** | | |
|  | Подготовка к приемке | Процедура |
|  | Проведение приемочных испытаний в соответствии с условиями задачи 5.1.1.9 | Протокол  (общий с результатом 5.3.13.1) |
|  | Управление конфигурацией поставленного ПП |  |

2.2.2. Процесс поставки (The Supply Process)

***Процесс поставки*** определяет работы и задачи поставщика. Процесс поставки начинается с решения о подготовке предложения в ответ на заявку на подряд, присланную заказчиком, или с подписания договора с заказчиком на поставку системы, ПП или программной услуги. Затем определяются процедуры и ресурсы, необходимые для управления и обеспечения проекта, включая разработку проектных планов и их выполнение.

Процесс поставки состоит из *семи работ* (рис. 6). Общее число задач по данным работам равно 23.

Табл. 2 содержит задачи, реализуемые при выполнении соответствующих работ процесса поставки, и типы выходных результатов данных задач, которые должны быть документально оформлены.

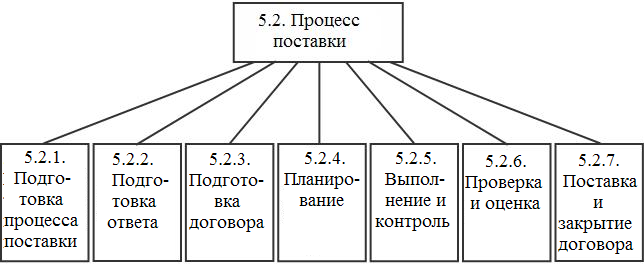


Рис. 6. Структура процесса поставки

Таблица 2

**Задачи работ процесса поставки**

|  |  |  |
| --- | --- | --- |
| № задачи в ИСО/МЭК 12207 | Описание задачи | Тип выходного результата  задачи |
| 5.2.1. Задачи работы ***Подготовка процесса поставки*** | | |
|  | Анализ требований в заявке на подряд |  |
|  | Принятие решения об участии в тендере или о подписании договора |  |
| 5.2.2. Задачи работы ***Подготовка ответа*** | | |
|  | Подготовка предложения в ответ на заявку о подряде, включая предложения по адаптации *СТБ ИСО/МЭК 12207-2003* | Предложение |
| 5.2.3. Задачи работы ***Подготовка договора*** | | |
|  | Проведение переговоров с заказчиком |  |
|  | Внесение изменений в текст договора по согласованию с заказчиком |  |
| 5.2.4. Задачи работы ***Планирование*** | | |
|  | Анализ требований к заказу |  |
|  | Определение модели жизненного цикла программных средств со структурированием в нее процессов, работ и задач *СТБ ИСО/МЭК 12207-2003* | Описание (в составе  результата задачи 5.2.4.5) |
|  | Определение требований к планированию управления и обеспечения проекта, обеспечения качества ПП или программной услуги, ресурсов | Описание (в составе  результата 5.2.4.5) |
|  | Анализ вариантов разработки программного продукта или предоставления программной услуги | Описание (в составе  результата 5.2.4.5) |
|  | Разработка и документальное оформление плана (планов) управления проектом | План |
| 5.2.5. Задачи работы ***Выполнение и контроль*** | | |
|  | Реализация планов управления проектом |  |
|  | Разработка, проведение опытной эксплуатации и сопровождение ПП в соответствии с процессами разработки, эксплуатации и сопровождения |  |
|  | Надзор за реализацией проекта, выявление и решение проблем |  |
|  | Управление и контроль деятельности субподрядчиков в соответствии с процессом заказа | Процедура |
|  | Взаимодействие с верифицирующей, аттестующей или проверяющей организацией |  |
|  | Взаимодействие с другими исполнителями договора |  |
| 5.2.6. Задачи работы ***Проверка и оценка*** | | |
|  | Координация работы по проверке выполнения договора |  |
|  | Участие в совещаниях, подготовке приемки ПП, приемочных испытаниях, совместных анализах и аудиторских проверках |  |
|  | Выполнение верификации и аттестации ПП, программной услуги и процессов |  |
|  | Предоставление заказчику отчетов об оценках, анализах, аудиторских проверках, испытаниях, решениях проблем |  |
|  | Обеспечение заказчику доступа к ресурсам для проверки ПП или программной услуги |  |
|  | Выполнение работ по обеспечению качества |  |
| 5.2.7. Задачи работы ***Поставка и закрытие договора*** | | |
|  | Поставка ПП или программной услуги заказчику |  |
|  | Помощь заказчику в поддержке поставленного ПП или программной услуги |  |

***План управления*** (см. задачу 5.2.4.5) должен содержать следующие вопросы:

* планирование организационной структуры проекта, полномочий и обязанностей каждого участника проекта;
* планирование технической среды разработки, эксплуатации и сопровождения; техническия среда включает условия проведения испытаний, оборудование, организация архивной библиотеки, средства, стандарты, процедуры, инструментарий;
* планирование структуры распределения заданий по процессам и работам жизненного цикла совместно со сметами, составом исполнителей, требуемыми материальными ресурсами, необходимыми программными средствами, графиками выполнения задач;
* планирование управления характеристиками качества создаваемого программного продукта или программной услуги;
* планирование управления безопасностью, защитой и другими критическими требованиями к программному продукту или программной услуге;
* планирование обеспечения защиты, включая правила доступа к информации на уровне каждой проектной организации;
* планирование управления критическими ситуациями (областями проекта, связанными с потенциальными техническими, финансовыми и плановыми затруднениями);
* планирование управления субподрядчиками;
* планирование обеспечения качества;
* планирование верификации и аттестации;
* планирование взаимоотношений с заказчиками, реализуемых совместными анализами, аудиторскими проверками, совещаниями, отчетами, модификациями и изменениями программного продукта, сдачей, утверждением, приемкой ПП и договорами;
* планирование взаимоотношений с пользователями, реализуемых посредством выполнения требуемых настроек, демонстраций прототипов и оценок;
* планирование подтверждения статуса поставляемой продукции (инструкции, обязательная сертификация, права собственности, использования и распространения, гарантии и лицензионные права);
* планирование средств для планирования, надзора и отчетности;
* планирование обучения персонала.

2.2.3. Процесс разработки (The Development Process)

***Процесс разработки*** определяет работы и задачи разработчика. Данный процесс включает работы по анализу требований, проектированию, программированию, сборке, тестированию, вводу в действие и приемке программного продукта или системы.

Процесс разработки состоит из *тринадцати работ* (). Общее число задач по данным работам равно 55.

Табл. 3 – 15 содержат задачи, реализуемые при выполнении соответствующих работ процесса разработки, и типы выходных результатов данных задач, которые должны быть документально оформлены.

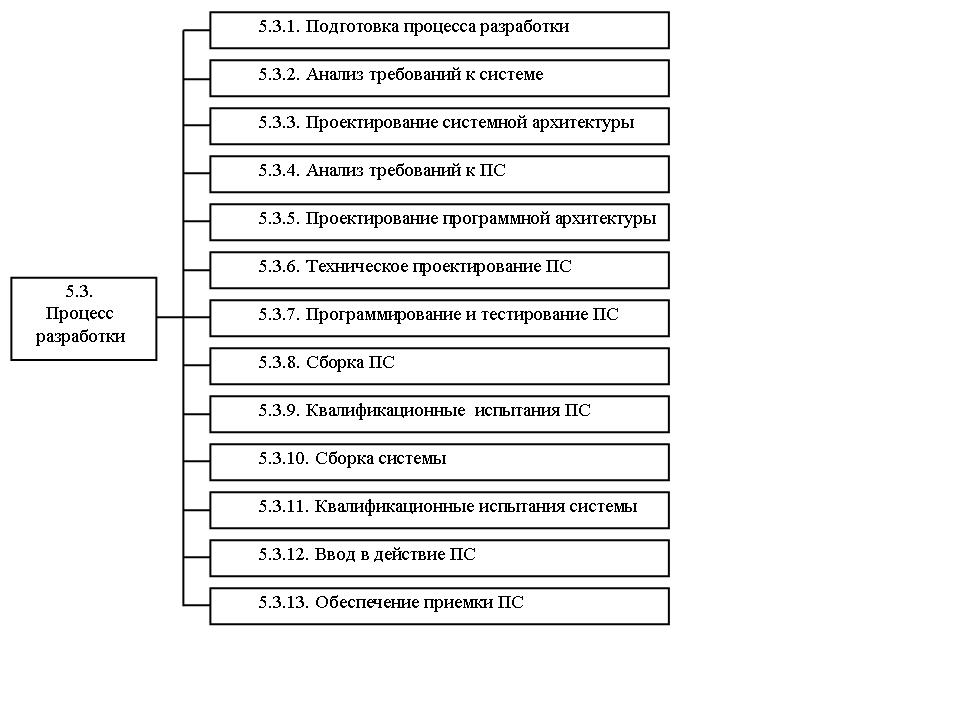


Рис. 7. Структура процесса разработки

Таблица 3

**Задачи работы Подготовка процесса разработки**

|  |  |  |
| --- | --- | --- |
| № задачи в ИСО/МЭК 12207 | Описание задачи | Тип выходного результата  задачи |
|  | Определение модели жизненного цикла программных средств со структурированием в нее процессов, работ и задач *СТБ ИСО/МЭК 12207-2003* (если модель не определена в договоре) | Описание (в составе  результата задачи 5.1.3.4) |
|  | Документальное оформление выходных результатов, управление конфигурацией выходных результатов, решение возникающих проблем, выполнение вспомогательных процессов в соответствии с процессами группы 6 (см. рис. 13) | Протоколы и отчеты в составе результатов соответствующих работ |
|  | Выбор и адаптация стандартов, методов, инструментариев, языков программирования | Описание (в составе  результата 5.1.3.4) |
|  | Разработка и выполнение планов проведения работ процесса разработки | План |
|  | Поставка всех комплектующих изделий |  |

Таблица 4

**Задачи работы Анализ требований к системе**

|  |  |  |
| --- | --- | --- |
| № задачи в ИСО/МЭК 12207 | Описание задачи | Тип выходного результата  задачи |
|  | Анализ области применения системы и определение требований к ней (рис. 8) | Описание |
|  | Оценка требований к системе по критериям (см. рис. 8) | Отчет |

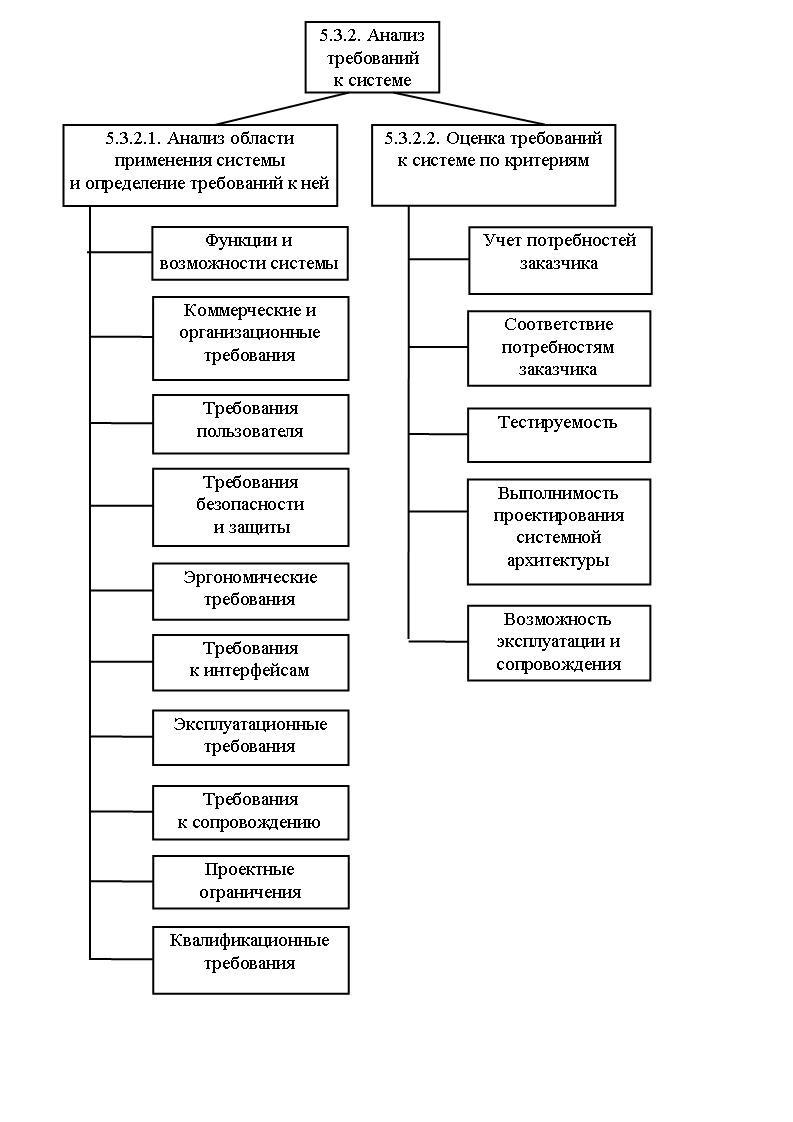


Рис. . Состав требований к системе и критерии их оценки

Таблица 5

**Задачи работы Проектирование системной архитектуры**

|  |  |  |
| --- | --- | --- |
| № задачи в ИСО/МЭК 12207 | Описание задачи | Тип выходного результата  задачи |
|  | Определение общей архитектуры системы, распределение требований к ней между объектами технических и программных средств архитектуры и ручными операциями и дальнейшее уточнение требований | Описание |
|  | Оценка архитектуры системы и требований к объектам архитектуры по критериям | Отчет |

При *оценке архитектуры системы и требований к ее объектам* (см. задачу в табл. 5) должны быть проанализированы следующие *критерии*:

* учет требований к системе;
* соответствие требованиям к системе;
* соответствие используемых стандартов и методов проектирования;
* возможность программных объектов архитектуры выполнять установленные для них требования;
* возможность эксплуатации и сопровождения.

Таблица 6

**Задачи работы Анализ требований к программным средствам**

|  |  |  |
| --- | --- | --- |
| № задачи в ИСО/МЭК 12207 | Описание задачи | Тип выходного результата  задачи |
|  | Определение требований к ПС () | Описание |
|  | Оценка требований к ПС по критериям (см. ) | Отчет |
|  | Проведение совместных анализов в соответствии с процессом совместного анализа | Протокол |

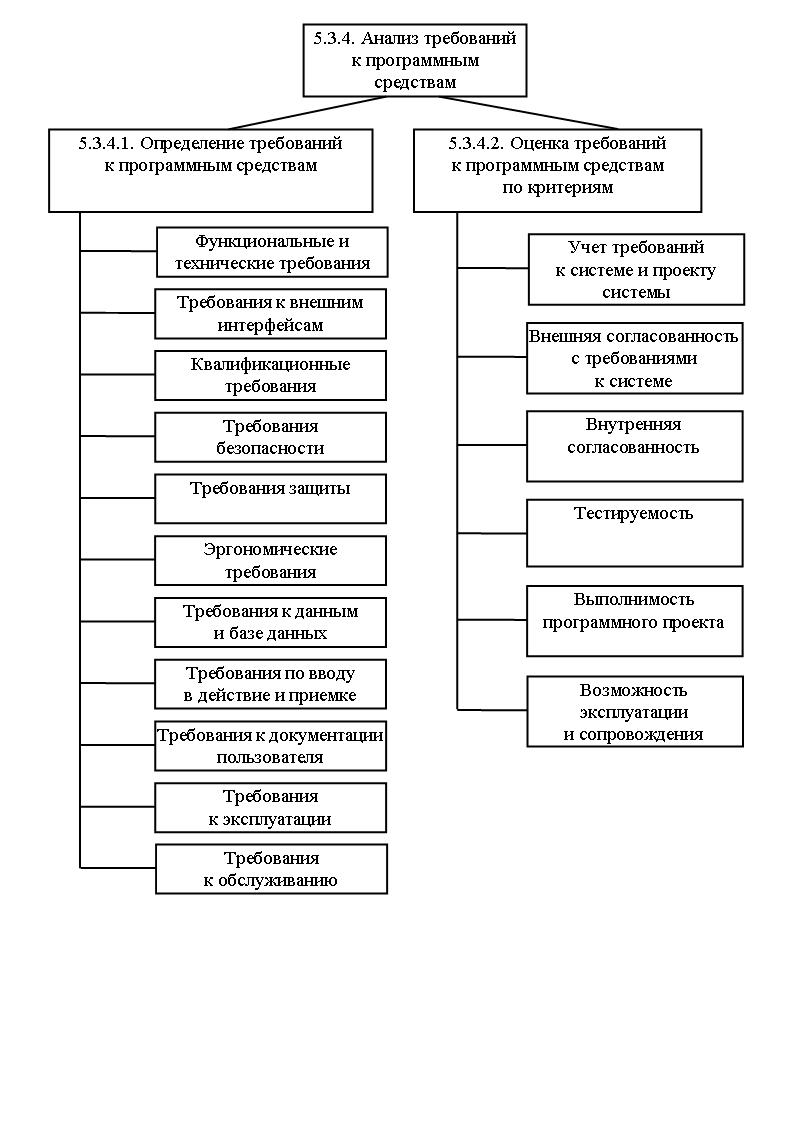


Рис. . Состав требований к программным средствам и критерии их оценки

Таблица 7

**Задачи работы Проектирование программной архитектуры**

|  |  |  |
| --- | --- | --- |
| № задачи в ИСО/МЭК 12207 | Описание задачи | Тип выходного результата  задачи |
|  | Преобразование требований к программному объекту в его архитектуру, распределение требований к программному объекту между его компонентами и дальнейшее уточнение требований (разработка эскизного проекта) | Описание |
|  | Разработка эскизного проекта интерфейсов программного объекта и его компонентов | Описание |
|  | Разработка эскизного проекта базы данных | Описание |
|  | Разработка предварительных версий документации пользователя | Руководство |
|  | Разработка предварительных требований к испытаниям (тестированию) программного объекта и графика сборки программного продукта | Описание, план |
|  | Оценка архитектуры программного объекта и эскизных проектов интерфейсов и базы данных по критериям | Отчет |
|  | Проведение совместных анализов в соответствии с процессом совместного анализа |  |

При *оценке архитектуры программного объекта и эскизных проектов интерфейсов и базы данных* (см. задачу ) должны быть учтены *критерии*:

* учет требований к программному объекту;
* внешняя согласованность с требованиями к программному объекту;
* внутренняя согласованность между компонентами объекта;
* соответствие методов проектирования и используемых стандартов;
* возможность технического проектирования;
* возможность эксплуатации и сопровождения.

Таблица 8

**Задачи работы Техническое проектирование программных средств**

|  |  |  |
| --- | --- | --- |
| № задачи в ИСО/МЭК 12207 | Описание задачи | Тип выходного результата  задачи |
|  | Разработка технического проекта для компонентов программного объекта (представление их в виде набора программных модулей), распределение технических требований к компонентам между программными модулями и дальнейшее уточнение требований | Описание |
|  | Разработка технического проекта интерфейсов программного объекта, его компонентов и модулей | Описание |
|  | Разработка технического проекта базы данных | Описание |
|  | Уточнение документации пользователя | Руководство |
|  | Разработка требований к испытаниям и программе испытаний программных модулей | Описание, план |
|  | Уточнение общих требований к испытанию (тестированию) и программе сборки программных средств | Описание, план |
|  | Оценка технического проекта и требований к тестированию по критериям | Отчет |
|  | Проведение совместных анализов в соответствии с процессом совместного анализа |  |

При *оценке технического проекта* *и требований к тестированию* (см. задачу ) должны быть учтены следующие *критерии*:

* учет требований к программному объекту;
* внешнее соответствие спроектированной архитектуре;
* внутренняя согласованность между компонентами программного объекта и программными модулями;
* соответствие методов проектирования и используемых стандартов;
* возможность тестирования;
* возможность эксплуатации и сопровождения.

Работы по проектированию программных средств (см. табл. 4 – табл. 8) поясняет рис. 10.

На данном рисунке отражена структура основных результатов работ процесса разработки, связанных с проектированием ПС, и взаимосвязь данных результатов.

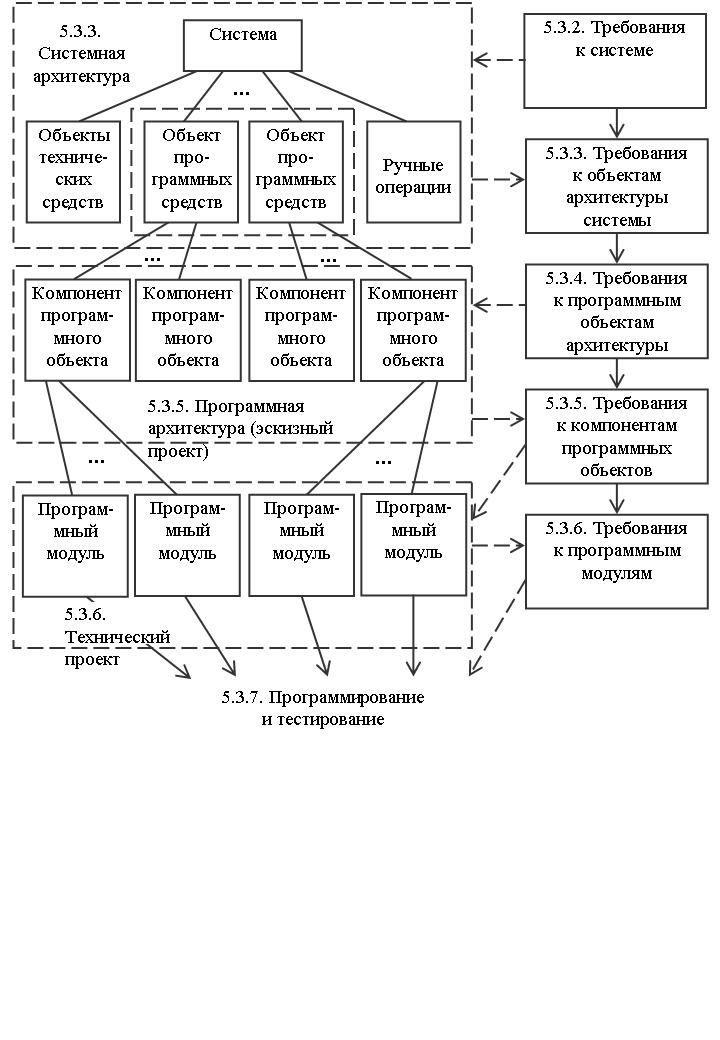


Рис. . Структура результатов работ по проектированию программных средств

Таблица 9

**Задачи работы Программирование и тестирование программных средств**

|  |  |  |
| --- | --- | --- |
| № задачи в ИСО/МЭК 12207 | Описание задачи | Тип выходного результата  задачи |
|  | Программирование программных модулей, базы данных, разработка процедур и данных для их испытаний (тестирования) | Программный продукт,  описание, процедура |
|  | Тестирование программных модулей и базы данных | Протокол |
|  | Уточнение документации пользователя | Руководство |
|  | Уточнение общих требований к тестированию и программы сборки программного средства | Описание, план |
|  | Оценка результатов программирования и тестирования по критериям | Отчет |

При *оценке результатов программирования и тестирования* (см. задачу ) должны быть учтены следующие *критерии*:

* учет требований к программному объекту и проекту объекта в целом;
* внешнее соответствие требованиям и проекту программного объекта;
* внутреннее соответствие между требованиями к программным модулям;
* тестовое покрытие всех модулей;
* соответствие методов программирования и используемых стандартов;
* возможность сборки и тестирования;
* возможность эксплуатации и сопровождения.

Таблица 10

**Задачи работы Сборка программных средств**

|  |  |  |
| --- | --- | --- |
| № задачи в ИСО/МЭК 12207 | Описание задачи | Тип выходного результата  задачи |
|  | Разработка плана сборки для объединения программных модулей и компонентов в программный объект | План |
|  | Сборка программных модулей и компонентов и тестирование результатов сборки | Описание, протокол |
|  | Уточнение документации пользователя | Руководство |
|  | Разработка плана и процедуры квалификационных испытаний программных средств | План, процедура |
|  | Оценка планов и результатов сборки и тестирования, а также документации пользователя по критериям | Отчет |
|  | Проведение совместного анализа в соответствии с процессом совместного анализа |  |

При *оценке планов и результатов сборки и тестирования, а также документации пользователя* (см. задачу ) должны быть учтены следующие *критерии*:

* учет требований к системе;
* внешнее соответствие требованиям к системе;
* внутренняя согласованность между программными объектами;
* тестовое покрытие требований к программному объекту;
* соответствие используемых стандартов и методов испытаний;
* соответствие ожидаемым результатам;
* выполнимость квалификационного испытания программного объекта;
* возможность эксплуатации и сопровождения.

Таблица 11

**Задачи работы Квалификационные испытания программных средств**

|  |  |  |
| --- | --- | --- |
| № задачи в ИСО/МЭК 12207 | Описание задачи | Тип выходного результата  задачи |
|  | Проведение квалификационных испытаний (тестирования) программных средств | Протокол |
|  | Уточнение документации пользователя | Руководство |
|  | Оценка результатов испытаний и документации пользователя по критериям | Отчет |
|  | Обеспечение проведения аудиторской проверки в соответствии с процессом аудита | Протокол |
|  | Доработка и подготовка программного продукта к последующим работам, определение состояния конфигурации (базовой линии) проекта | Протокол |

При *оценке результатов испытаний и документации пользователя* (см. задачу ) должны быть учтены следующие *критерии*:

* тестовое покрытие требований к программному объекту;
* соответствие ожидаемым результатам;
* возможность сборки и тестирования системы;
* возможность эксплуатации и сопровождения.

Таблица 12

**Задачи работы Сборка системы**

|  |  |  |
| --- | --- | --- |
| № задачи в ИСО/МЭК 12207 | Описание задачи | Тип выходного результата  задачи |
|  | Сборка объектов программной и технической конфигурации, ручных операций, других систем в единую систему, испытания собранной системы | Описание, протокол |
|  | Разработка плана и процедуры квалификационных испытаний системы | План, процедура |
|  | Оценка собранной системы по критериям | Отчет |

При *оценке собранной системы* (см. задачу ) должны быть учтены следующие *критерии*:

* тестовое покрытие требований к системе;
* соответствие методов тестирования и используемых стандартов;
* соответствие ожидаемым результатам;
* возможность квалификационных испытаний системы;
* возможность эксплуатации и сопровождения.

Таблица 13

**Задачи работы Квалификационные испытания системы**

|  |  |  |
| --- | --- | --- |
| № задачи в ИСО/МЭК 12207 | Описание задачи | Тип выходного результата  задачи |
|  | Проведение квалификационных испытаний системы | Описание, протокол |
|  | Оценка системы по критериям | Отчет |
|  | Обеспечение проведения аудиторской проверки в соответствии с процессом аудита | Протокол |
|  | Доработка и подготовка программного продукта к последующим работам, определение состояния конфигурации (базовой линии) проекта | Протокол |

При *оценке системы* (см. задачу ) должны быть учтены следующие *критерии*:

* тестовое покрытие требований к системе;
* соответствие ожидаемым результатам;
* возможность эксплуатации и сопровождения.

Таблица 14

**Задачи работы Ввод в действие программных средств**

|  |  |  |
| --- | --- | --- |
| № задачи в ИСО/МЭК 12207 | Описание задачи | Тип выходного результата  задачи |
|  | Разработка плана ввода в действие программного продукта в среде эксплуатации | План |
|  | Ввод в действие программного продукта в соответствии с планом | Протокол |

Таблица 15

**Задачи работы Обеспечение приемки программных средств**

|  |  |  |
| --- | --- | --- |
| № задачи в ИСО/МЭК 12207 | Описание задачи | Тип выходного результата  задачи |
|  | Обеспечение проведения заказчиком приемочных испытаний | Процедура, протокол |
|  | Укомплектование и поставка программного продукта заказчику на условиях договора | Программный продукт |
|  | Обеспечение обучения и поддержка персонала заказчика на условиях договора |  |

2.2.4. Процесс эксплуатации (The Operation Process)

***Процесс эксплуатации*** определяет работы и задачи оператора. Данный процесс включает эксплуатацию программного продукта и поддержку пользователей в процессе эксплуатации. Процесс эксплуатации состоит из *четырех работ* (). Общее число задач по данным работам равно 9.

Табл. 16 содержит задачи, реализуемые при выполнении соответствующих работ процесса эксплуатации, и типы выходных результатов данных задач, которые должны быть документально оформлены.

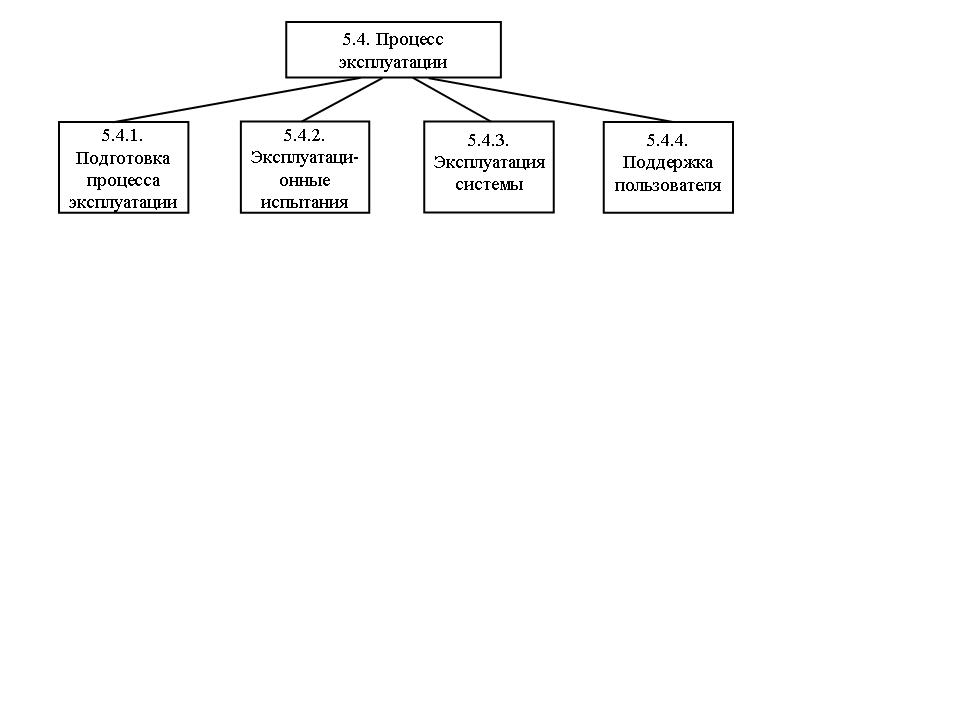


Рис. . Структура процесса эксплуатации

Таблица 16

**Задачи работ процесса эксплуатации**

|  |  |  |
| --- | --- | --- |
| № задачи в ИСО/МЭК 12207 | Описание задачи | Тип выходного результата  задачи |
| 5.4.1. Задачи работы ***Подготовка процесса эксплуатации*** | | |
|  | Разработка плана эксплуатации и определение набора стандартов по эксплуатации | План |
|  | Установка процедур документирования и решения проблем | Процедура |
|  | Установка процедур для тестирования ПП в эксплуатационной среде, ввода сообщений о проблеме в процесс сопровождения и ввода ПП в эксплуатацию | Процедура |
| 5.4.2. Задачи работы ***Эксплуатационные испытания*** | | |
|  | Проведение эксплуатационных испытаний и ввод программных продуктов в промышленную эксплуатацию | Протокол |
|  | Обеспечение инициализации и эксплуатации программ и базы данных в соответствии с планом эксплуатации |  |
| 5.4.3. Задачи работы ***Эксплуатация системы*** | | |
|  | Эксплуатация в установленной среде в соответствии с документацией пользователя |  |
| 5.4.4. Задачи работы ***Поддержка пользователя*** | | |
|  | Помощь и консультации пользователям в установленном порядке | Протокол |
|  | Передача запросов пользователя в процесс сопровождения и контроль их решения | Протокол |
|  | Обеспечение временного решения проблем |  |

2.2.5. Процесс сопровождения (The Maintenance Process)

***Процесс сопровождения*** определяет работы и задачи персонала сопровождения и реализуется при модификациях программного продукта. Цель процесса – изменение существующего ПП при сохранении его целостности. Процесс охватывает вопросы переносимости и снятия ПП с эксплуатации.

Процесс сопровождения состоит из *шести работ* (рис. 12).

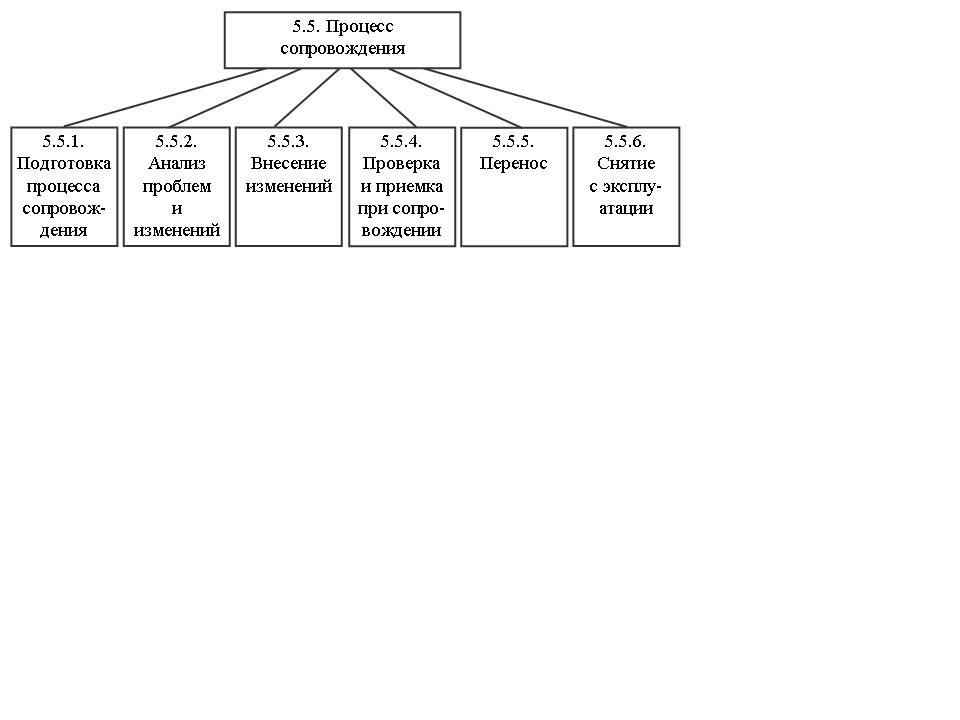


Рис. . Структура процесса сопровождения

Общее число задач по данным работам равно 24.

Табл. 17 содержит задачи, реализуемые при выполнении соответствующих работ процесса сопровождения, и типы выходных результатов данных задач, которые должны быть документально оформлены.

Таблица 17

**Задачи работ процесса сопровождения**

|  |  |  |
| --- | --- | --- |
| № задачи в ИСО/МЭК 12207 | Описание задачи | Тип выходного результата  задачи |
| 5.5.1. Задачи работы ***Подготовка процесса сопровождения*** | | |
|  | Разработка планов и процедур для проведения работ по сопровождению | План, процедура |
|  | Определение процедур для документирования возникающих проблем и организации связи с процессом решения проблем | Процедура |
|  | Реализация процесса управления конфигурацией для управления изменениями существующей системы |  |
| 5.5.2. Задачи работы ***Анализ проблем и изменений*** | | |
|  | Анализ сообщений о проблеме |  |
|  | Дублирование или верификация проблемы |  |
|  | Разработка вариантов реализации изменения |  |
|  | Документальное оформление сообщения о проблеме и вариантов ее решения | Протокол, отчет |
|  | Согласование выбранного варианта изменения в соответствии с договором |  |
| 5.5.3. Задачи работы ***Внесение изменений*** | | |
|  | Определение документов и программных модулей, требующих изменения | Протокол |
|  | Реализация изменений с использованием процесса разработки, разработка процедур испытаний и оценки результатов испытаний | Процедура, протокол,  отчет |
| 5.5.4. Задачи работы ***Проверка и приемка при сопровождении*** | | |
|  | Совместная проверка работоспособности измененной системы |  |
|  | Приемка внесенного изменения | Протокол |
| 5.5.5. Задачи работы ***Перенос*** | | |
|  | Обеспечение соответствия измененного при переносе в новую эксплуатационную среду программного продукта или данных стандарту *СТБ ИСО/МЭК 12207-2003* |  |
|  | Разработка и выполнение плана переноса программного продукта | План |
|  | Уведомление пользователей о планах и работах по переносу программного продукта | Отчет |
|  | Обеспечение параллельной эксплуатации в прежней и новой среде (при необходимости) |  |
|  | Уведомление заинтересованных сторон о выполненном переносе, передача в архив документации по прежней среде |  |
|  | Итоговый анализ влияния переноса на эксплуатацию системы и рассылка результатов анализа заинтересованным сторонам | Отчет |
|  | Обеспечение доступности информации и документации, связанных с прежней средой |  |
| 5.5.6. Задачи работы ***Снятие с эксплуатации*** | | |
|  | Разработка плана снятия с эксплуатации | План |
|  | Уведомление пользователей о планах и работах по снятию с эксплуатации | Отчет |
|  | Проведение параллельной эксплуатации прежнего и нового программных продуктов |  |
|  | Уведомление заинтересованных сторон о снятии ПП с эксплуатации и передача документации о нем в архив |  |
|  | Обеспечение доступности информации по снятому с эксплуатации ПП |  |

### 2.3. Вспомогательные процессы жизненного цикла

Вспомогательные процессы жизненного цикла ПС и систем представлены на рис. 13. На данном рисунке сохранена нумерация процессов, принятая в стандарте *СТБ ИСО/МЭК 12207-2003*.

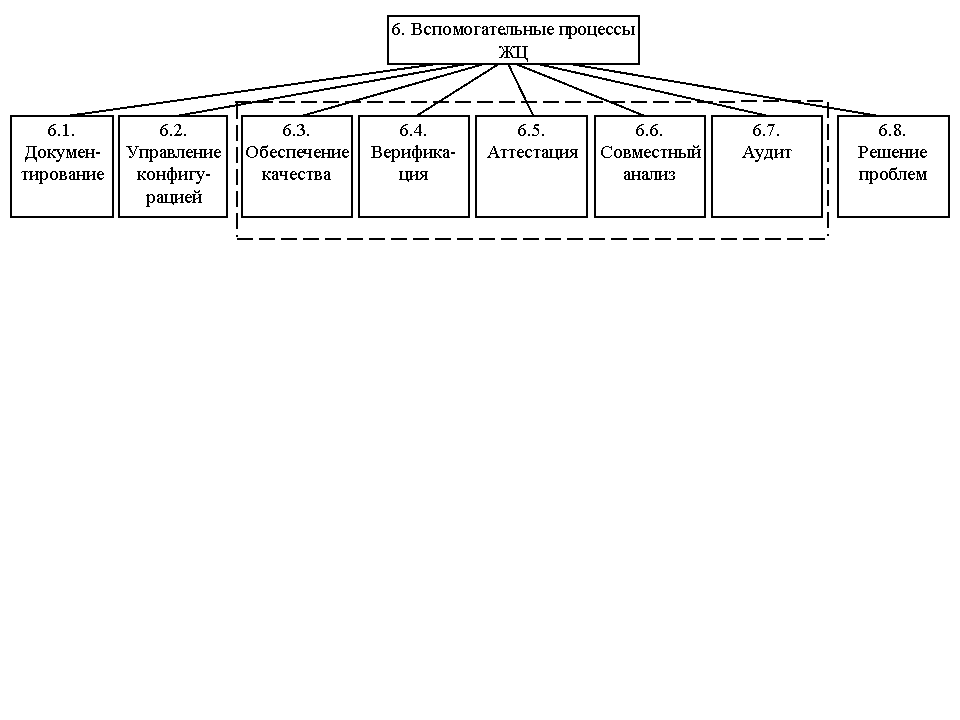


Рис. . Вспомогательные процессы жизненного цикла программных средств и систем

2.3.1. Процесс документирования (The Documentation Process)

***Процесс документирования*** является процессом формализованного описания информации, созданной в процессе или работе жизненного цикла. Он включает планирование, проектирование, разработку, выпуск, редактирование, распространение и сопровождение документов по программному продукту.

Процесс документирования состоит из *четырех работ* (рис. 14). Общее число задач по данным работам равно 7.

Табл. 18 содержит задачи, реализуемые при выполнении соответствующих работ процесса документирования, и типы выходных результатов данных задач, которые должны быть документально оформлены.

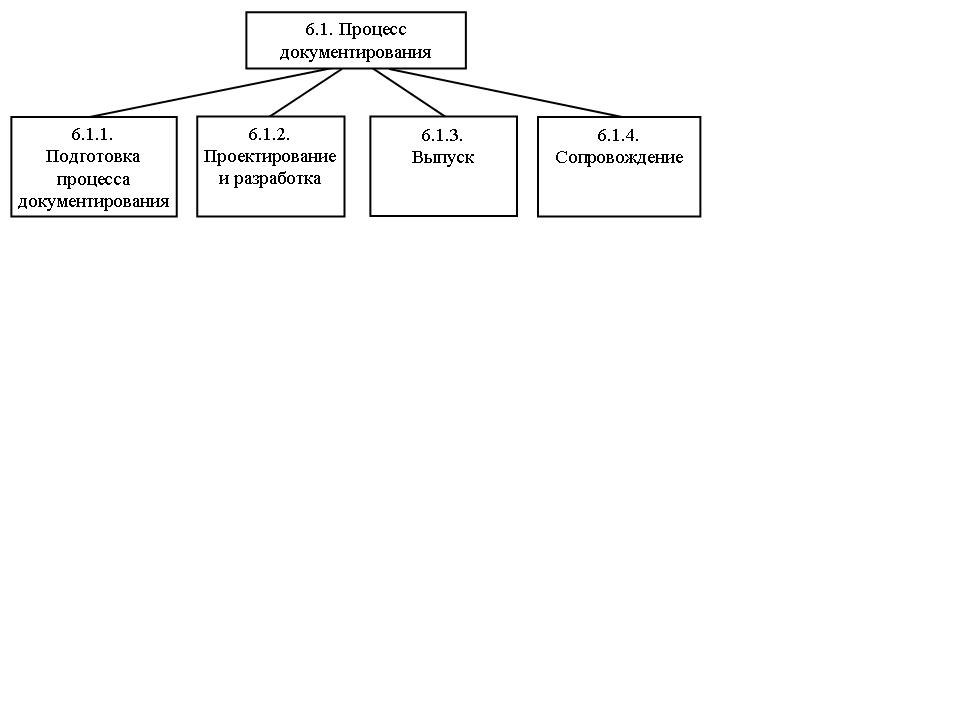


Рис. . Структура процесса документирования

Таблица 18

Задачи работ процесса документирования

|  |  |  |
| --- | --- | --- |
| № задачи в ИСО/МЭК 12207 | Описание задачи | Тип выходного результата  задачи |
| 6.1.1. Задачи работы ***Подготовка процесса документирования*** | | |
|  | Разработка и реализация плана обозначения документов, выпускаемых в процессах ЖЦ ПП | План |
| 6.1.2. Задачи работы ***Проектирование и разработка*** | | |
|  | Проектирование документов согласно стандартам на документацию | Описание |
|  | Подтверждение источника и соответствия исходных материалов для документов |  |
|  | Проверка и редактирование документов согласно стандартам, утверждение компетентными лицами |  |
| 6.1.3. Задачи работы ***Выпуск*** | | |
|  | Издание и распространение документов в соответствии с планом |  |
|  | Управление документированием в соответствии с процессом управления конфигурацией |  |
| 6.1.4. Задачи работы ***Сопровождение*** | | |
|  | Внесение изменений в документацию согласно процессам сопровождения и управления конфигурацией |  |

В ***плане обозначения документов*** (см. задачу ) должны быть определены:

* заголовок или наименование;
* назначение;
* пользователи документа;
* процедуры и обязанности по подготовке исходных материалов, разработке, проверке, изменению, утверждению, выпуску, хранению, распространению, сопровождению и управлению конфигурацией документов.

2.3.2. Процесс управления конфигурацией (The Configuration Management Process)

***Процесс управления конфигурацией*** является процессом применения административных и технических процедур на всем протяжении ЖЦ ПС для определения состояния (базовой линии) программных объектов в системе, управления их изменениями и выпуском.

Данный процесс состоит из *шести работ* (). Общее число задач по данным работам равно 6.

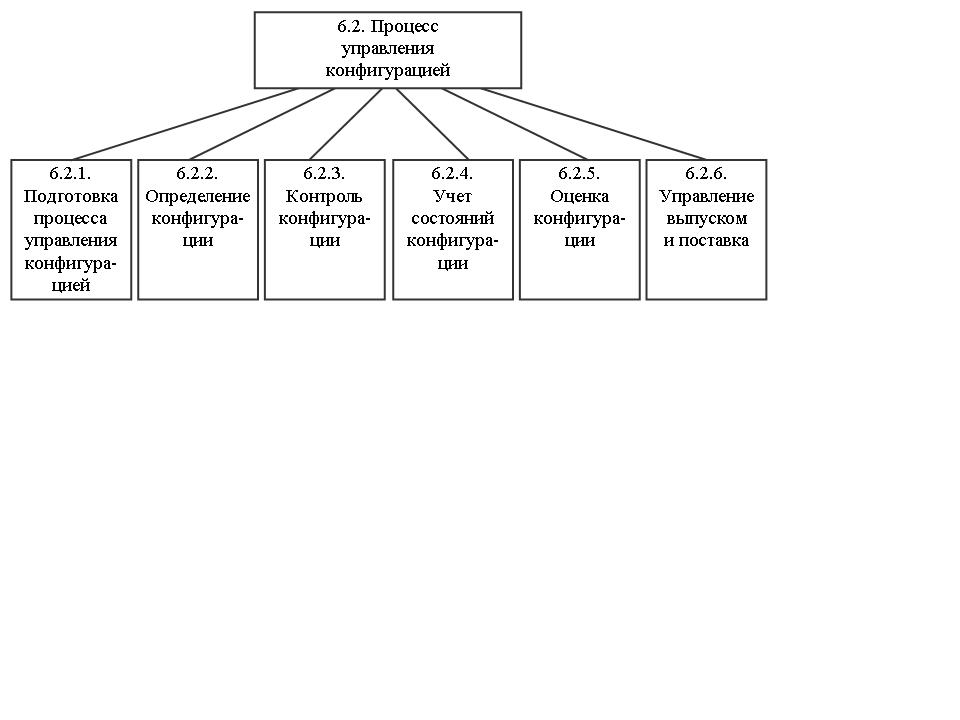


Рис. . Структура процесса управления конфигурацией

Табл. 19 содержит задачи, реализуемые при выполнении соответствующих работ процесса управления конфигурацией, и типы выходных результатов данных задач, которые должны быть документально оформлены.

Таблица 19

**Задачи работ процесса управления конфигурацией**

|  |  |  |
| --- | --- | --- |
| № задачи в ИСО/МЭК 12207 | Описание задачи | Тип выходного результата  задачи |
| 6.2.1. Задачи работы ***Подготовка процесса управления конфигурацией*** | | |
|  | Разработка плана управления конфигурацией | План |
| 6.2.2. Задачи работы ***Определение*** ***конфигурации*** | | |
|  | Определение схемы обозначения программных объектов и их версий (объектов программной конфигурации) и документации, в которой фиксируется состояние их конфигурации | Описание |
| 6.2.3. Задачи работы ***Контроль конфигурации*** | | |
|  | Регистрация заявок на внесение изменений; анализ и оценка изменений; принятие или непринятие заявки; реализация, верификация и выпуск измененного программного объекта; обеспечение аудиторских проверок изменений |  |
| 6.2.4. Задачи работы ***Учет состояний конфигурации*** | | |
|  | Подготовка протоколов управления конфигурацией и отчетов о состоянии контролируемых программных объектов | Протокол, отчет |
| 6.2.5. Задачи работы ***Оценка*** ***конфигурации*** | | |
|  | Определение и обеспечение функциональной законченности и физической завершенности программных объектов | Протокол, отчет |
| 6.2.6. Задачи работы ***Управление выпуском и поставка*** | | |
|  | Контроль выпуска и поставки программных продуктов и документации |  |

2.3.3. Процесс обеспечения качества (The Quality Assurance Process)

***Процесс обеспечения качества*** является процессом обеспечения гарантий того, что программные продукты и процессы в жизненном цикле проекта соответствуют требованиям и планам. Данный процесс должен быть независимым от субъектов, участвующих в проекте. Это позволяет достичь объективности процесса. При обеспечении качества могут использоваться результаты процессов верификации, аттестации, совместного анализа, аудита и решения проблем.

Процесс обеспечения качества состоит из *четырех работ* (). Общее число задач по данным работам равно 16.

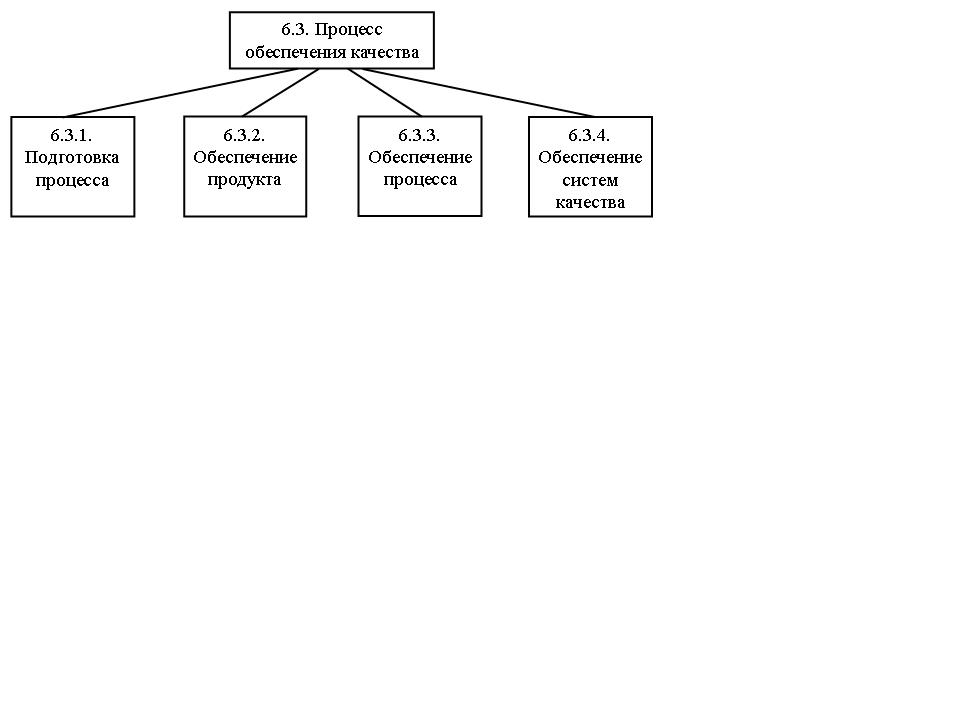


Рис. . Структура процесса обеспечения качества

Табл. 20 содержит задачи, реализуемые при выполнении соответствующих работ процесса обеспечения качества, и типы выходных результатов данных задач, которые должны быть документально оформлены.

Таблица 20

**Задачи работ процесса обеспечения качества**

|  |  |  |
| --- | --- | --- |
| № задачи в ИСО/МЭК 12207 | Описание задачи | Тип выходного результата  задачи |
| 6.3.1. Задачи работы ***Подготовка процесса обеспечения качества*** | | |
|  | Адаптация процесса обеспечения качества к условиям конкретного проекта, определение целей процесса обеспечения качества | Описание (в составе  результата задачи 6.3.1.3) |
|  | Координация с процессами верификации, аттестации, совместного анализа и аудита |  |
|  | Разработка и реализация плана обеспечения качества | План |
|  | Выполнение запланированных и традиционных работ и задач по обеспечению качества, при необходимости организация связи с процессом решения проблем | Отчет, протокол |
|  | Предоставление отчетов о работах и задачах по обеспечению качества заказчику согласно договору |  |
|  | Обеспечение организационной независимости лиц, реализующих работы и задачи по обеспечению качества |  |
| 6.3.2. Задачи работы ***Обеспечение продукта*** | | |
|  | Обеспечение соответствия условиям договора и выполнения всех планов |  |
|  | Обеспечение соответствия программных продуктов и документации условиям договора и планам |  |
|  | Обеспечение соответствия поставляемых программных продуктов требованиям, установленным в договоре |  |
| 6.3.3. Задачи работы ***Обеспечение процесса*** | | |
|  | Обеспечение соответствия процессов жизненного цикла программных средств условиям договора и утвержденным планам |  |
|  | Обеспечение соответствия технологий программирования, условий разработки, условий испытаний и архивных библиотек условиям договора |  |
|  | Обеспечение соответствия программных продуктов, разработанных субподрядчиком, требованиям основного договора |  |
|  | Обеспечение взаимной поддержки заказчика и других участников договора в соответствии с условиями договора и планами |  |
|  | Обеспечение соответствия характеристик программных продуктов и процессов установленным стандартам и процедурам |  |
|  | Обеспечение соответствия и обучения исполнителей проекта |  |
| 6.3.4. Задачи работы ***Обеспечение систем качества*** | | |
|  | Обеспечение проведения дополнительных работ по управлению качеством в соответствии с разделами *СТБ ИСО 9001*, указанными в договоре | Протокол |

2.3.4. Процесс верификации (The Verification Process)

Стандарт *СТБ ИСО/МЭК 12207-2003* дает такое определение процесса верификации.

***Процесс верификации*** является процессом определения того, что программные продукты функционируют в полном соответствии с требованиями и условиями, реализованными в предшествующих работах.

Данное определение является не совсем понятным для лиц, некомпетентных в области верификации. Кроме того, оно сужает применимость верификации только до программных продуктов. Для пояснения данного определения следует рассмотреть само понятие верификации. Термин *верификация* *(verification)* обозначает подтверждение с помощью экспертизы и представления объективных доказательств того, что конкретные требования полностью реализованы. В процессе разработки верификация связана с экспертизой результатов данной работы с целью определения их соответствия установленным на входе данной работы требованиям [, ]. Таким образом, верификация может применяться не только к программным продуктам, но и к любым другим результатам работы. Например, верификации могут подвергаться требования, системная и программная архитектура, документация, методы, планы и т.п.

В этой связи более понятным и близким к истине является следующее определение процесса верификации. ***Процесс верификации*** – это процесс определения того, что результаты работы соответствуют требованиям или условиям, установленным на входе данной работы.

Процесс верификации может включать анализ, проверку и тестирование. Объектами анализа и проверки могут являться, например, документация и исходные тексты программных модулей. Объектами тестирования могут являться исполнимые коды программных модулей, компонентов, промежуточных и конечного программных продуктов.

Данный процесс называется ***процессом независимой верификации***, если организация–исполнитель не зависит от поставщика, разработчика, оператора или персонала сопровождения.

Процесс верификации состоит из *двух работ* (). Общее число задач по данным работам равно 13.

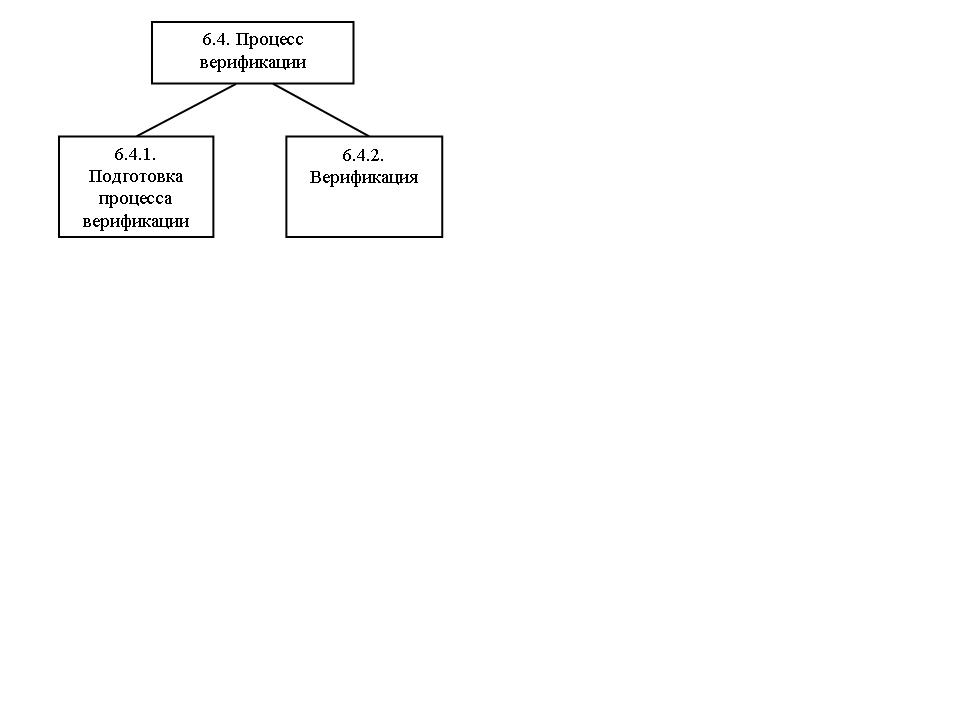


Рис. . Структура процесса верификации

Табл. 21 содержит задачи, реализуемые при выполнении соответствующих работ процесса верификации, и типы выходных результатов данных задач, которые должны быть документально оформлены.

Таблица 21

**Задачи работ процесса верификации**

|  |  |  |
| --- | --- | --- |
| № задачи в ИСО/МЭК 12207 | Описание задачи | Тип выходного результата задачи |
| 6.4.1. Задачи работы ***Подготовка процесса верификации*** | | |
|  | Определение необходимости в проекте работ по верификации и степени их организационной независимости, анализ критичности проектных требований | Описание (в составе  результата задачи 6.4.1.5) |
|  | Установка процесса верификации (при необходимости) | Описание (в составе результата 6.4.1.5) |
|  | Выбор соответствующей независимой квалифицированной организации (при необходимости) | Описание (в составе результата 6.4.1.5) |
|  | Определение верифицируемых работ и продуктов, выбор работ и задач верификации | Описание (в составе результата 6.4.1.5) |
|  | Разработка плана верификации | План |
|  | Реализация плана проведения верификации, обеспечение доступности его результатов заказчику, при необходимости организация связи с процессом решения проблем | Отчет, протокол |
| 6.4.2. Задачи работы ***Верификация*** | | |
|  | Верификация договора по критериям |  |
|  | Верификация процесса по критериям |  |
|  | Верификация требований по критериям |  |
|  | Верификация результатов проектирования по критериям |  |
|  | Верификация исходных текстов программных модулей по критериям |  |
|  | Верификация сборки по критериям |  |
|  | Верификация документации по критериям |  |

***Договор*** должен быть верифицирован (см. задачу ) по следующим *критериям*:

* возможность поставщика удовлетворять установленным требованиям;
* непротиворечивость требований и охват ими потребностей пользователя;
* наличие соответствующих процедур для внесения изменений в установленные требования и для решения проблем;
* наличие процедур по взаимодействию и кооперации между участниками договора;
* наличие критериев и процедур, предусмотренных в соответствии с установленными требованиями.

***Процесс*** должен быть верифицирован (см. задачу ) по следующим *критериям*:

* соответствие и своевременность установления требований к планированию проекта;
* пригодность, реализуемость, выполнимость в соответствии с планом и условиями договора выбранных для проекта процессов;
* применимость стандартов, процедур и условий к процессам проекта;
* укомплектованность и обученность персонала в соответствии с условиями договора.

***Требования*** должны быть верифицированы (см. задачу ) по следующим *критериям*:

* непротиворечивость, выполнимость и тестируемость требований к системе;
* распределение требований к системе между объектами технических и программных средств и ручных операций в соответствии с критериями проектирования;
* непротиворечивость, выполнимость, тестируемость и точность отражения требований к системе в требованиях к программным средствам;
* правильность (подтвержденная соответствующими методами) критических требований к программным средствам, в том числе по безопасности и защите.

***Результаты проектирования*** должны быть верифицированы (см. задачу ) по следующим *критериям*:

* правильность, соответствие установленным требованиям и учет этих требований;
* реализация соответствующей последовательности событий, исходных данных, выходных результатов, интерфейсов, логики; соответствие временным ограничениям и ограничениям размера; обнаружение, локализация и устранение ошибок;
* возможность дальнейшего использования с учетом требований;
* правильность, подтвержденная соответствующими методами, реализации требований безопасности, защиты и других критических требований.

***Исходные тексты программных модулей*** должны быть верифицированы (см. задачу ) по следующим *критериям*:

* трассируемость с результатами проектирования и требованиями; тестируемость, правильность и соответствие требованиям и стандартам программирования;
* реализация соответствующей последовательности событий, соответствующих интерфейсов, правильных данных и логики управления; завершенность; соответствие временным ограничениям и ограничениям размера; обнаружение, локализация и устранение ошибок;
* соответствие результатам проектирования и требованиям;
* возможность дальнейшего использования с учетом результатов тестирования и требований;
* правильность, подтвержденная соответствующими методами, реализации требований безопасности, защиты и других критических требований.

***Сборка*** должна быть верифицирована (см. задачу ) по следующим *критериям*:

* полнота и правильность сборки программных компонентов и модулей каждого программного объекта в соответствующий программный объект;
* полнота и правильность сборки технических и программных объектов и ручных операций в систему;
* выполнение задач сборки в соответствии с планом сборки.

***Документация*** должна быть верифицирована (см. задачу ) по следующим *критериям*:

* соответствие, полнота и непротиворечивость документации;
* своевременность подготовки документации;
* соблюдение установленных процедур управления конфигурацией документов.

2.3.5. Процесс аттестации (The Validation Process)

В стандарте *СТБ ИСО/МЭК 12207-2003* дано такое определение процесса аттестации.

***Процесс аттестации*** является процессом определения полноты соответствия установленных требований, созданной системы или программного продукта их функциональному назначению.

Чтобы пояснить это определение, следует рассмотреть понятие аттестации.

Термин *аттестация (validation)* обозначает подтверждение экспертизой и представлением объективных доказательств того, что конкретные требования к конкретным объектам полностью реализованы. В процессе разработки аттестациясвязана с экспертизой продукта в целях определения его соответствия потребностям пользователя (то есть *исходным требованиям к проекту*) [, ].

Аттестации (проверке на соответствие исходным требованиям) могут подвергаться любые промежуточные продукты процесса разработки. Аттестация может проводиться на начальных этапах работы и как часть работы по обеспечению приемки ПС.

Данный процесс называется процессом ***независимой аттестации***, если организация–исполнитель не зависит от поставщика, разработчика, оператора или персонала сопровождения.

Процесс аттестации состоит из *двух работ* (). Общее число задач по данным работам равно 10.

Табл. 22 содержит задачи, реализуемые при выполнении соответствующих работ процесса аттестации, и типы выходных результатов данных задач, которые должны быть документально оформлены.

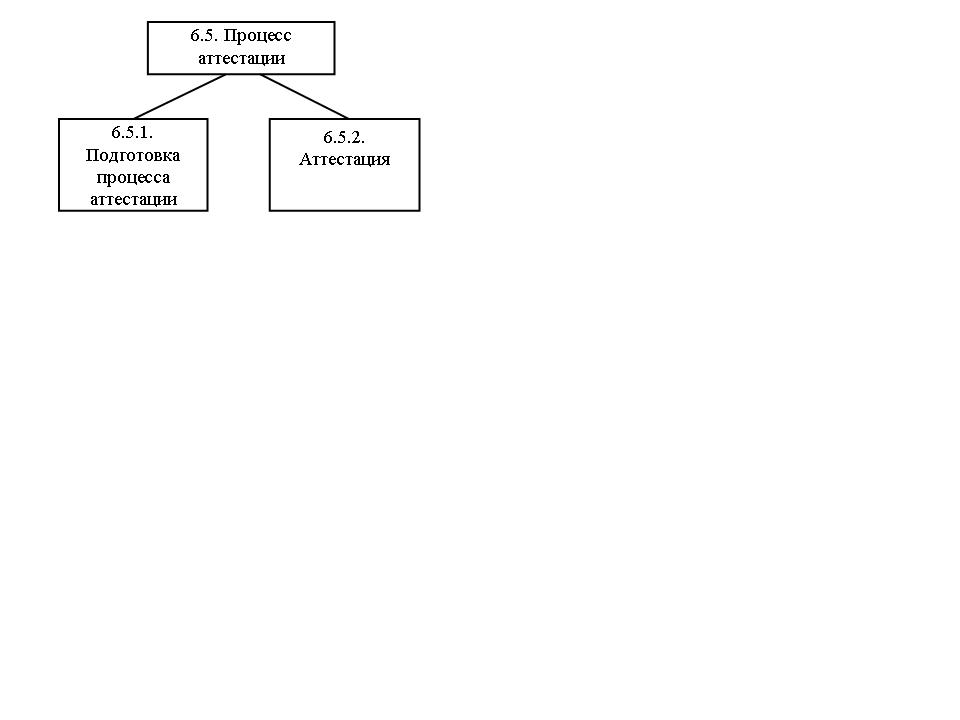


Рис. . Структура процесса аттестации

Таблица 22

**Задачи работ процесса аттестации**

|  |  |  |
| --- | --- | --- |
| № задачи в ИСО/МЭК 12207 | Описание задачи | Тип выходного результата  задачи |
| 6.5.1. Задачи работы ***Подготовка процесса аттестации*** | | |
|  | Определение необходимости в проекте работ по аттестации и степени их организационной независимости |  |
|  | Установка процесса, задач, методов, методик и средств аттестации (при необходимости) | Описание (в составе  результата задачи 6.5.1.4) |
|  | Выбор соответствующей независимой квалифицированной организации (при необходимости) |  |
|  | Разработка плана аттестации | План |
|  | Реализация плана проведения аттестации, обеспечение доступности его результатов заказчику, при необходимости организация связи с процессом решения проблем (см. п. пособия) | Отчет |
| 6.5.2. Задачи работы ***Аттестация*** | | |
|  | Подготовка выбранных требований к испытаниям (тестированию), контрольных примеров и технических условий испытаний к анализу результатов испытаний | Описание |
|  | Обеспечение соответствия требований к испытаниям (тестированию), контрольных примеров и технических условий испытаний требованиям к объектам аттестации |  |
|  | Проведение испытаний | Протокол |
|  | Подтверждение соответствия программного продукта заданным возможностям | Протокол |
|  | Проведение испытаний программного продукта в заданном контексте использования | Протокол |

2.3.6. Процесс совместного анализа (The Joint Review Process)

***Процесс совместного анализа*** является процессом оценки состояний и результатов работ по проекту. Совместные анализы проводятся в течение всего договора и применяются как на уровне управления проектом, так и на уровне его технической реализации. Данный процесс может выполняться двумя любыми сторонами, участвующими в договоре, когда одна сторона (анализирующая) проверяет другую (анализируемую).

Процесс совместного анализа состоит из *трех работ* (). Общее число задач по данным работам равно 8.

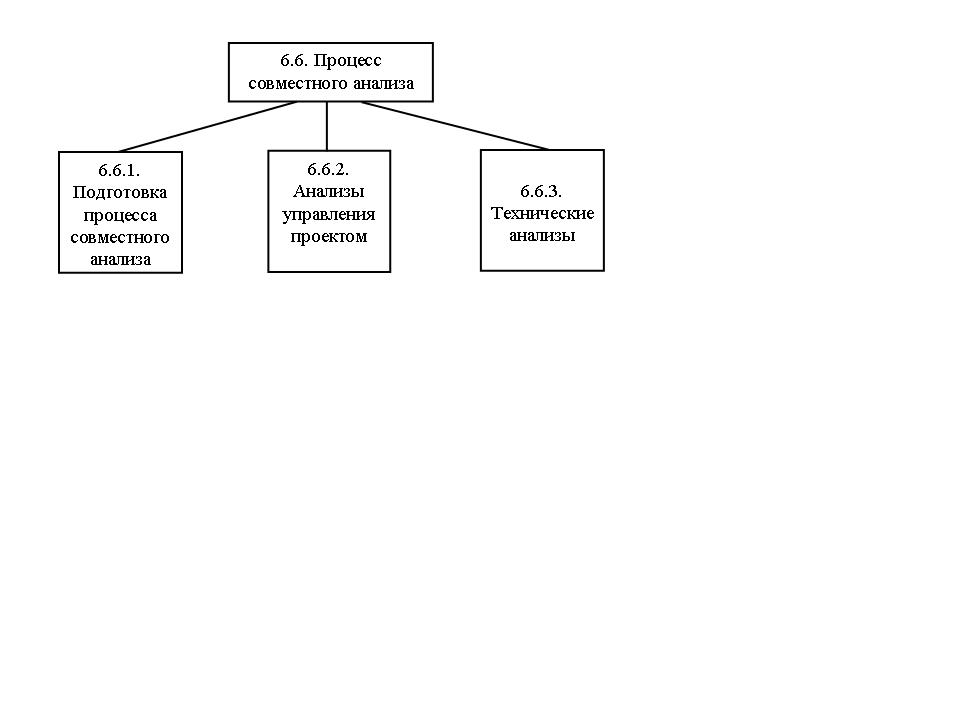


Рис. . Структура процесса совместного анализа

Табл. 23 содержит задачи, реализуемые при выполнении соответствующих работ процесса совместного анализа, и типы выходных результатов данных задач, которые должны быть документально оформлены.

Таблица 23

Задачи работ процесса совместного анализа

|  |  |  |
| --- | --- | --- |
| № задачи в ИСО/МЭК 12207 | Описание задачи | Тип выходного результата  задачи |
| 6.6.1. Задачи работы ***Подготовка процесса совместного анализа*** | | |
|  | Организация проведения периодических и целевых анализов хода работ в установленные сроки |  |
|  | Согласование ресурсов, необходимых для проведения анализа, между участвующими в нем сторонами | Протокол |
|  | Согласование плана, объема, процедур и критериев проведения анализа, состава анализируемых результатов работ | Протокол |
|  | Передача выявленных при проведении анализа проблем в процесс решения проблем |  |
|  | Передача результатов анализа заинтересованным сторонам |  |
|  | Согласование результатов анализа, принимаемых обязательств и критериев завершения анализа | Протокол |
| 6.6.2. Задачи работы ***Анализы управления проектом*** | | |
|  | Оценка соответствия состояния проекта проектным планам, графикам, стандартам и руководствам | Отчет, протокол |
| 6.6.3. Задачи работы ***Технические анализы*** | | |
|  | Оценка соответствия состояния создаваемых программных продуктов или услуг проектным планам, графикам, стандартам и руководствам | Отчет |

2.3.7. Процесс аудита (The Audit Process)

***Процесс аудита*** является процессом определения соответствия требованиям, планам и условиям договора. Данный процесс может выполняться двумя сторонами, участвующими в договоре, когда одна сторона (ревизующая) проверяет другую сторону (ревизуемую).

Процесс аудита состоит из *двух работ* (). Общее число задач по данным работам равно 8.

Табл. 24 содержит задачи, реализуемые при выполнении соответствующих работ процесса аудита, и типы выходных результатов данных задач, которые должны быть документально оформлены.

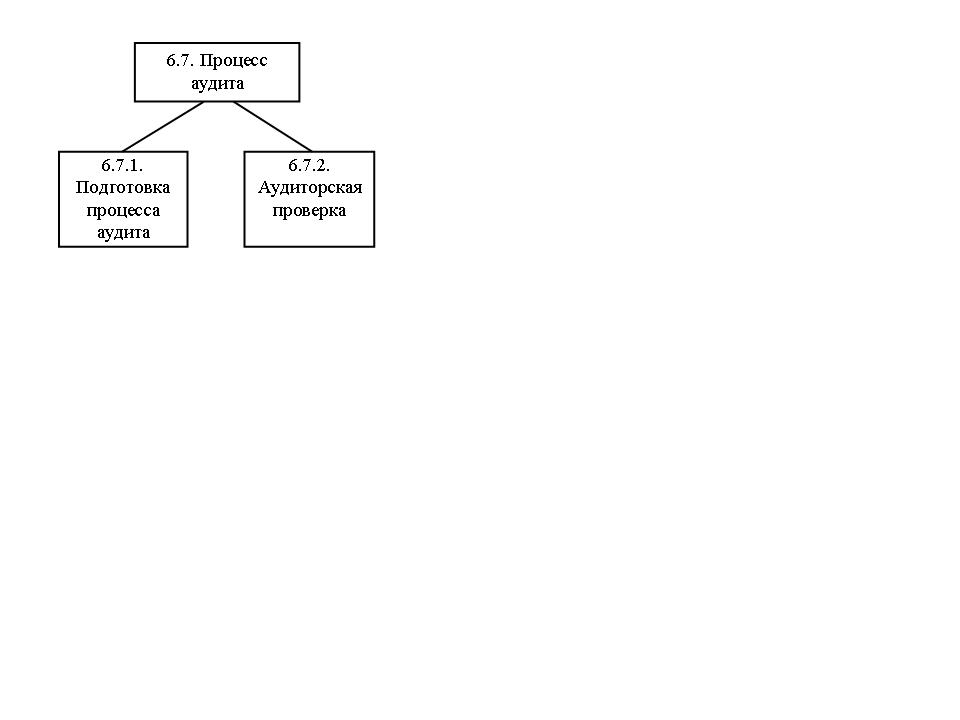


Рис. . Структура процесса аудита

Таблица 24

Задачи работ процесса аудита

| № задачи в ИСО/МЭК 12207 | Описание задачи | Тип выходного результата  задачи |
| --- | --- | --- |
| 6.7.1. Задачи работы ***Подготовка процесса аудита*** | | |
|  | Организация проведения аудиторских проверок в сроки согласно плану |  |
|  | Обеспечение независимости аудиторского персонала |  |
|  | Согласование ресурсов, необходимых для проведения аудита, между участвующими в нем сторонами | Протокол |
|  | Согласование плана, объема, процедур и критериев проведения аудиторской проверки, состава проверяемых результатов работ | Протокол |
|  | Передача выявленных при проведении аудиторской проверки проблем в процесс решения проблем | Протокол |
|  | Передача результатов аудиторской проверки заинтересованным сторонам |  |
|  | Согласование результатов, принимаемых обязательств и критериев завершения аудиторской проверки | Протокол |
| 6.7.2. Задачи работы ***Аудиторская проверка*** | | |
|  | Проведение аудиторской проверки по критериям | Отчет |

*Критерии* *проведения* *аудиторской проверки* (см. задачу ):

* соответствие запрограммированных программных продуктов (объектов) проектной документации;
* пригодность подготовки приемки и требований к тестированию, установленных в документации, для приемки программных продуктов (объектов);
* соответствие тестовых данных установленным техническим требованиям;
* успешность тестирования программных продуктов и их соответствие установленным к ним требованиям;
* правильность отчетов об испытаниях (тестировании), устранение расхождений между фактическими и ожидаемыми результатами;
* соответствие документации пользователя установленным стандартам;
* выполнение работ в соответствии с утвержденными требованиями, планами и договором;
* соответствие стоимости и графика проведения работ утвержденным планам.

2.3.8. Процесс решения проблем (The Problem Resolution Process)

***Процесс решения проблем*** является процессом анализа и решения проблем (включая обнаруженные несоответствия), которые обнаружены в ходе выполнения разработки, эксплуатации, сопровождения или других процессов.

Данный процесс состоит из *двух работ* (). Общее число задач по данным работам равно 2.

Табл. 25 содержит задачи, реализуемые при выполнении соответствующих работ процесса решения проблем, и типы выходных результатов данных задач, которые должны быть документально оформлены.

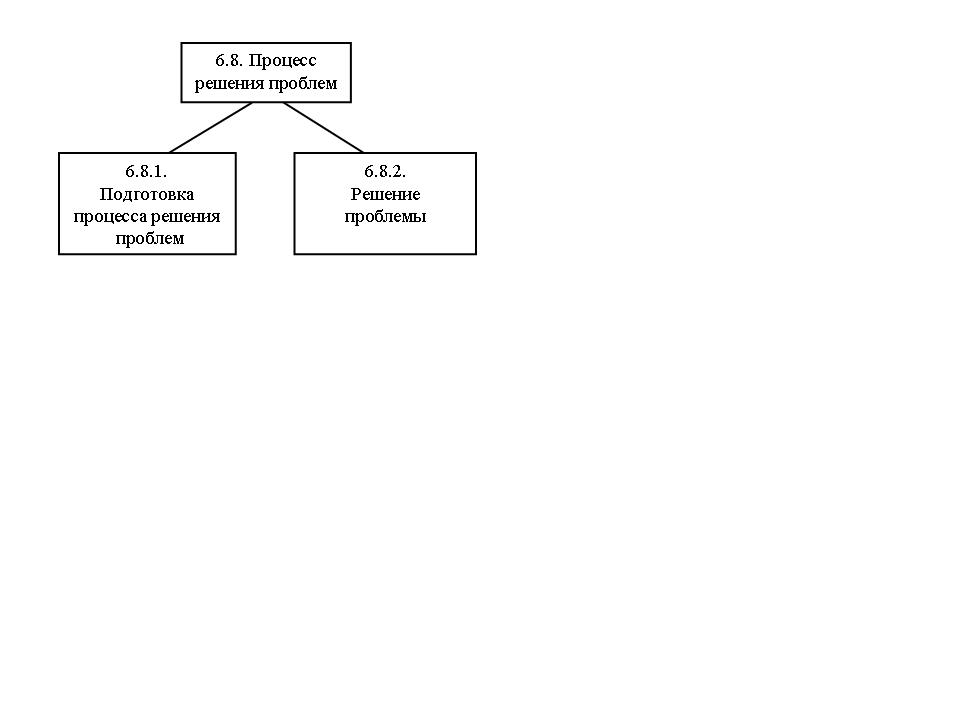


Рис. . Структура процесса решения проблем

Таблица 25

Задачи работ процесса решения проблем

|  |  |  |
| --- | --- | --- |
| № задачи в ИСО/МЭК 12207 | Описание задачи | Тип выходного результата  задачи |
| 6.8.1. Задачи работы ***Подготовка процесса решения проблем*** | | |
|  | Установка процесса решения проблем для обработки всех проблем, выявленных в программных продуктах и работах | Протокол |
| 6.8.2. Задачи работы ***Решение проблемы*** | | |
|  | Отслеживание проблем путем их выявления, исследования, анализа и решения и их документирование | Отчет |

### 2.4. Организационные процессы жизненного цикла

Организационные процессы жизненного цикла программных средств и систем представлены на . На данном рисунке сохранена нумерация процессов, принятая в *СТБ ИСО/МЭК 12207-2003*.

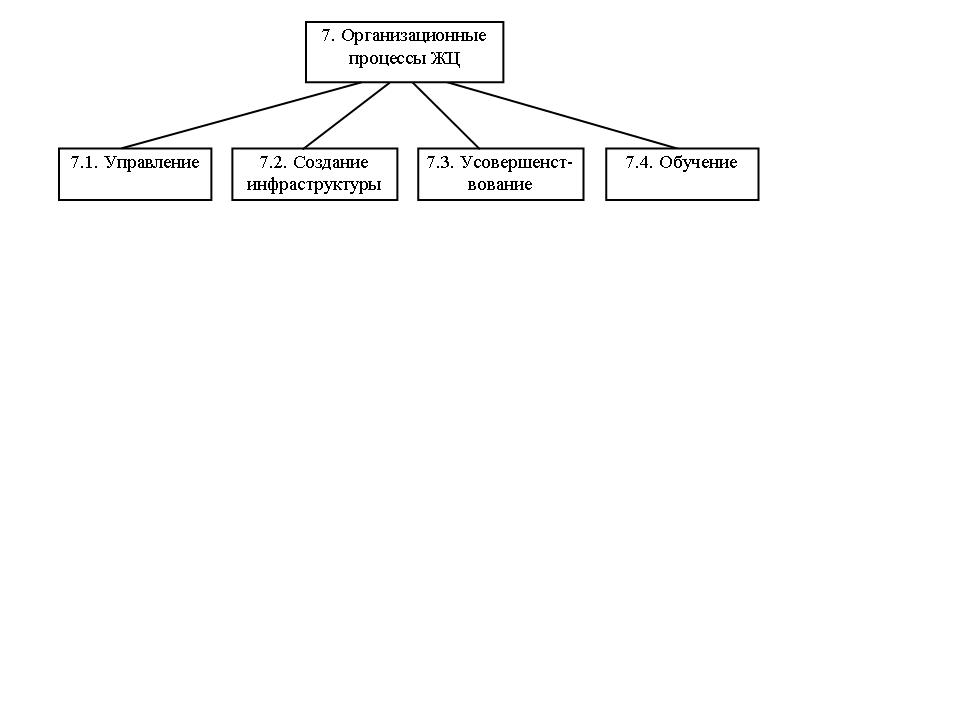


Рис. . Организационные процессы жизненного цикла программных средств и систем

2.4.1. Процесс управления (The Management Process)

***Процесс управления*** состоит из общих работ и задач, которые могут быть использованы любой стороной, управляющей соответствующим процессом. За управление продуктом, проектом, работами и задачами основных и вспомогательных процессов отвечает *администратор*.

Процесс управления состоит из *пяти работ* (). Общее число задач по данным работам равно 12.

Табл. 26 содержит задачи, реализуемые при выполнении соответствующих работ процесса управления, и типы выходных результатов данных задач, которые должны быть документально оформлены.

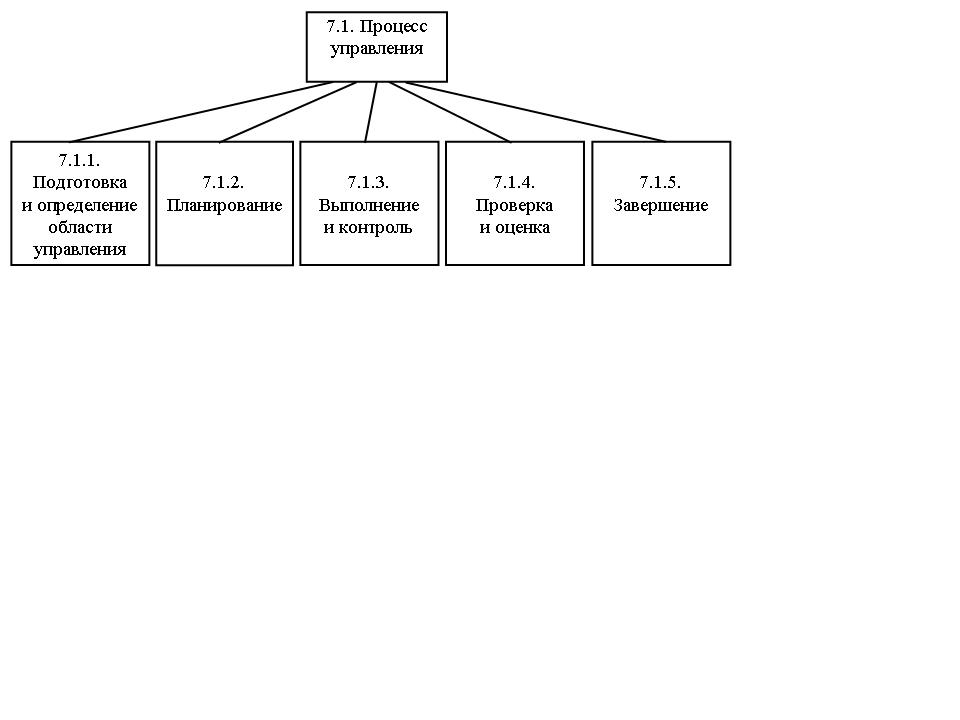


Рис. . Структура процесса управления

Таблица 26

**Задачи работ процесса управления**

|  |  |  |
| --- | --- | --- |
| № задачи в ИСО/МЭК 12207 | Описание задачи | Тип выходного результата  задачи |
| 7.1.1. Задачи работы ***Подготовка и определение области управления*** | | |
|  | Установка требований к реализуемому процессу | Описание |
|  | Определение возможности реализации процесса, проверка соответствия ресурсов и сроков | Описание |
|  | Изменение требований к процессу для удовлетворения критериев его завершения (при необходимости) | Протокол |
| 7.1.2. Задачи работы ***Планирование*** | | |
|  | Разработка планов выполнения процессов | План |
| 7.1.3. Задачи работы ***Выполнение и контроль*** | | |
|  | Управление процессом на базе разработанного плана |  |
|  | Текущий надзор за выполнением процесса | Отчет (в составе  результата задачи 7.1.3.4) |
|  | Исследование и решение проблем, обнаруженных при выполнении процесса, при необходимости изменение планов | Отчет (в составе  результата задачи 7.1.3.4) |
|  | Отчет в установленные сроки о реализации процесса | Отчет |
| 7.1.4. Задачи работы ***Проверка и оценка*** | | |
|  | Обеспечение оценки программных продуктов и планов на соответствие требованиям |  |
|  | Проверка результатов оценок программных продуктов, работ и задач на соответствие целям и планам | Отчет |
| 7.1.5. Задачи работы ***Завершение*** | | |
|  | Оценка всех программных продуктов, работ и задач по критериям, установленным в договоре | Отчет |
|  | Контроль результатов и полноты документации созданных программных продуктов и выполненных работ, передача их в архив |  |

***Планы выполнения процессов*** разрабатываются администратором проекта в ходе решения задачи 7.1.2.1. Они должны охватывать следующие вопросы:

* установка графиков выполнения задач;
* оценка необходимых трудозатрат;
* определение ресурсов, необходимых для выполнения задач;
* распределение задач по исполнителям;
* определение обязанностей исполнителей;
* определение критических ситуаций, связанных с задачами или самим процессом;
* установка используемых в процессе критериев управления качеством;
* определение затрат, связанных с реализацией процесса;
* обеспечение условий и определение инфраструктуры выполнения процесса.

2.4.2. Процесс создания инфраструктуры (The Infrastructure Process)

***Процесс создания инфраструктуры*** является процессом установления и сопровождения инфраструктуры, необходимой для любого другого процесса. *Инфраструктура* содержит технические и программные средства, инструментальные средства, методики, стандарты и условия для разработки, эксплуатации или сопровождения.

Процесс создания инфраструктуры состоит из *трех работ* (). Общее число задач по данным работам равно 5.

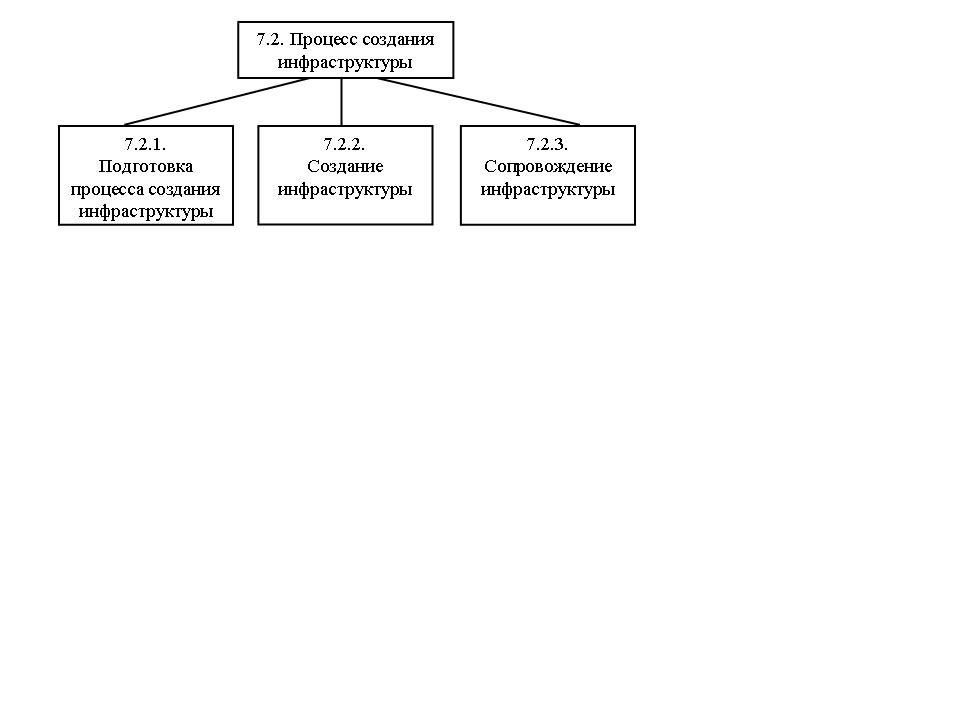


Рис. . Структура процесса создания инфраструктуры

Табл. 27 содержит задачи, реализуемые при выполнении соответствующих работ процесса создания инфраструктуры, и типы выходных результатов данных задач, которые должны быть документально оформлены.

Таблица 27

**Задачи работ процесса создания инфраструктуры**

|  |  |  |
| --- | --- | --- |
| № задачи в ИСО/МЭК 12207 | Описание задачи | Тип выходного результата  задачи |
| 7.2.1. Задачи работы ***Подготовка процесса создания инфраструктуры*** | | |
|  | Определение инфраструктуры, удовлетворяющей требованиям к процессу | Описание (в составе  результата задачи 7.2.1.2) |
|  | Разработка плана создания установленной инфраструктуры | План |
| 7.2.2. Задачи работы ***Создание инфраструктуры*** | | |
|  | Разработка конфигурации инфраструктуры с учетом характеристик продукта и процесса | Описание |
|  | Создание инфраструктуры к требуемому сроку |  |
| 7.2.3. Задачи работы ***Сопровождение* *инфраструктуры*** | | |
|  | Сопровождение, контроль и изменение инфраструктуры для удовлетворения требований к процессу, обеспечение связи с процессом управления конфигурацией | Описание |

2.4.3. Процесс усовершенствования (The Improvement Process)

***Процесс усовершенствования*** является процессом установления, оценки, измерения, контроля и улучшения любого процесса жизненного цикла программных средств.

Процесс усовершенствования состоит из *трех работ* (рис. 25).

Общее число задач по данным работам равно 6.

Табл. 28 содержит задачи, реализуемые при выполнении соответствующих работ процесса усовершенствования, и типы выходных результатов данных задач, которые должны быть документально оформлены.

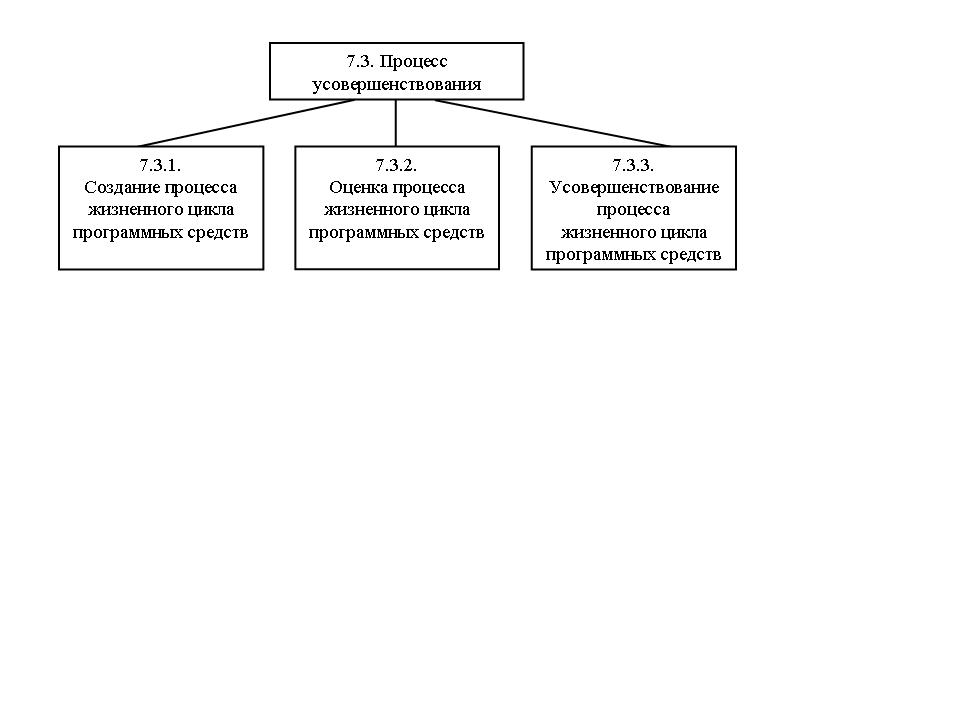


Рис. . Структура процесса усовершенствования

Таблица 28

Задачи работ процесса усовершенствования

|  |  |  |
| --- | --- | --- |
| № задачи в ИСО/МЭК 12207 | Описание задачи | Тип выходного результата задачи |
| 7.3.1. Задачи работы ***Создание* *процесса*** | | |
|  | Определение набора организационных процессов для всех процессов ЖЦ ПС и механизма управления процессом усовершенствования | Процедура |
| 7.3.2. Задачи работы ***Оценка* *процесса*** | | |
|  | Разработка и применение процедуры оценки процесса | Процедура |
|  | Планирование и выполнение анализов процессов в установленные сроки | План, отчет |
| 7.3.3. Задачи работы ***Усовершенствование* *процесса*** | | |
|  | Усовершенствование процесса (при необходимости) | Отчет |
|  | Сбор и анализ архивных, технических и оценочных данных по выполняемым процессам для их усовершенствования |  |
|  | Сбор, обновление и использование данных о расходах для усовершенствования организационных процессов административной деятельности |  |

2.4.4. Процесс обучения (The Training Process)

***Процесс обучения*** является процессом обеспечения первоначального и продолженного обучения персонала работам по заказу, поставке, разработке, эксплуатации или сопровождению программного проекта.

Процесс обучения состоит из *трех работ* (). Общее число задач по данным работам равно 4.

Табл. 29 содержит задачи, реализуемые при выполнении соответствующих работ процесса обучения, и типы выходных результатов данных задач, которые должны быть документально оформлены.

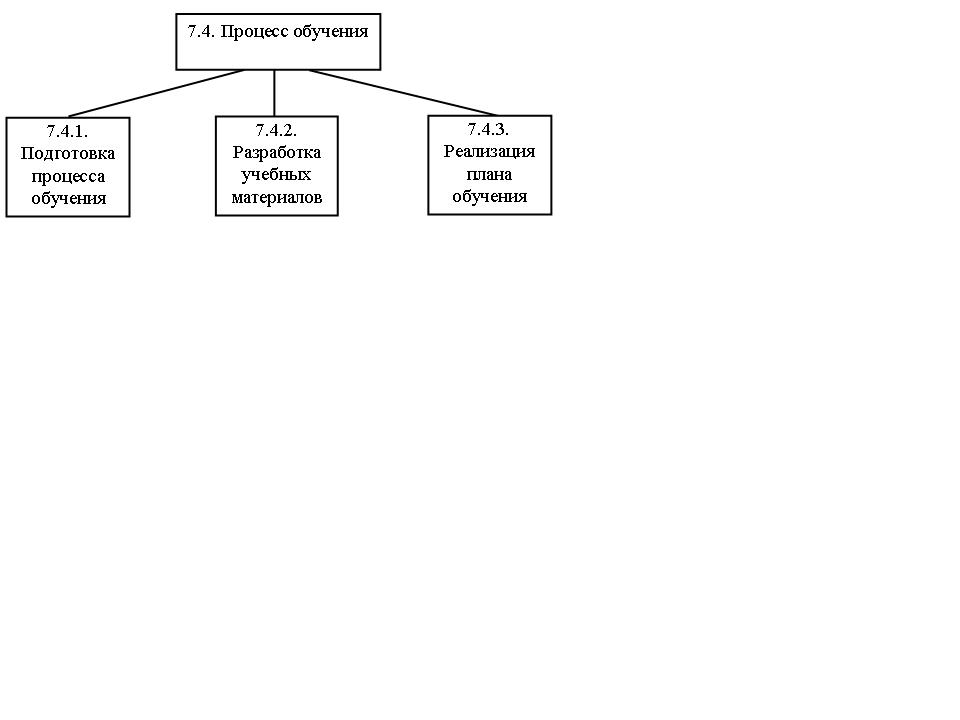


Рис. . Структура процесса обучения

Таблица 29

Задачи работ процесса обучения

| № задачи в ИСО/МЭК 12207 | Описание задачи | Тип выходного результата  задачи |
| --- | --- | --- |
| 7.4.1. Задачи работы ***Подготовка*** ***процесса обучения*** | | |
|  | Анализ требований к проекту для определения необходимости обучения персонала, разработка плана обучения | План |
| 7.4.2. Задачи работы ***Разработка учебных материалов*** | | |
|  | Разработка руководств и материалов для обучения | Описание |
| 7.4.3. Задачи работы ***Реализация плана обучения*** | | |
|  | Реализация плана обучения с сохранением протоколов | Протокол |
|  | Обеспечение своевременной подготовки персонала к выполнению запланированных работ и задач |  |

### 2.5. Адаптация требований стандарта СТБ ИСО/МЭК 12207-2003 к условиям проекта

Стандарт *СТБ ИСО/МЭК 12207-2003* определяет общий случай разработки типового проекта. В данном стандарте регламентируется *общее число процессов* жизненного цикла программных средств, равное 17, *общее число работ*, равное 74, *общее число задач*, равное 232.

Однако при разработке конкретных проектов может отсутствовать необходимость в использовании тех или иных процессов, работ или задач ЖЦ ПС. Кроме того, может потребоваться ввести в ЖЦ дополнительные процессы, работы и задачи, не регламентированные стандартом *СТБ ИСО/МЭК 12207-2003*. С учетом этого в интересах сокращения стоимости и улучшения качества конкретного проекта требования данного стандарта следует адаптировать. Процесс адаптации определен в *Приложении А СТБ ИСО/МЭК 12207-2003*.

***Процесс адаптации*** является процессом применения положений данного стандарта к условиям реализации конкретного программного проекта. Данный процесс следует относить к *дополнительным процессам жизненного цикла*.

Процесс адаптации состоит из *четырех работ* (). Номера данных работ соответствуют принятым в стандарте *СТБ ИСО/МЭК 12207-2003*. Общее число задач по работам равно 6.

Табл. 30 содержит задачи, реализуемые при выполнении соответствующих работ процесса адаптации, и типы выходных результатов данных задач, которые должны быть документально оформлены.

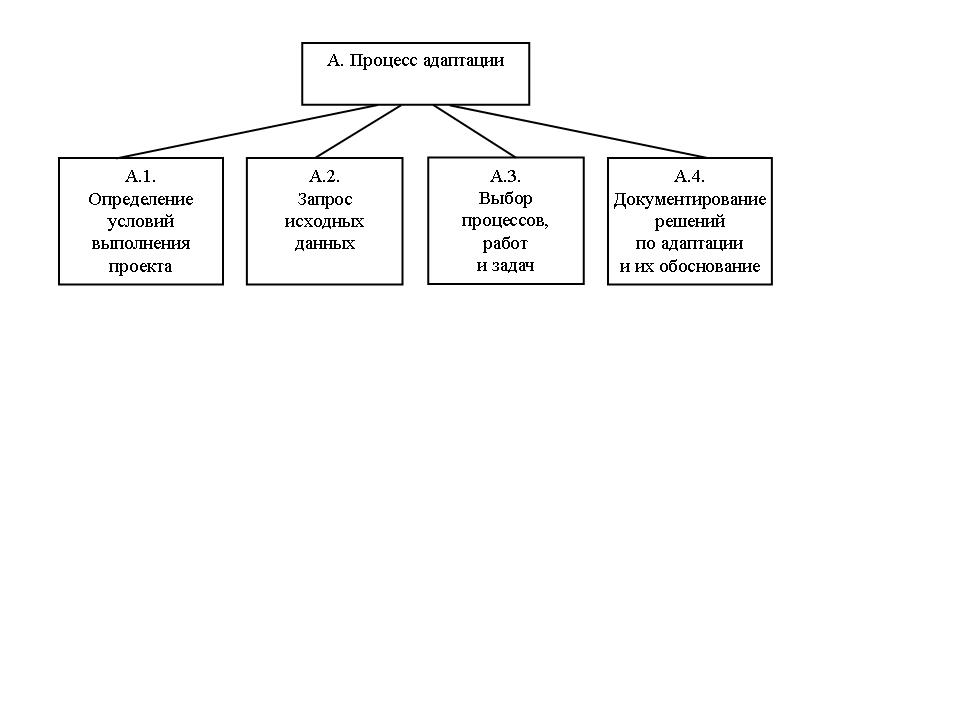


Рис. . Структура процесса адаптации

Таблица 30

**Задачи работ процесса адаптации**

| № задачи в ИСО/МЭК 12207 | Описание задачи | Тип выходного результата  задачи |
| --- | --- | --- |
| А.1. Задачи работы ***Определение условий выполнения проекта*** | | |
| А.1.1 | Определение характеристик условий выполнения проекта, влияющих на адаптацию |  |
| А.2. Задачи работы ***Запрос исходных данных*** | | |
| А.2.1 | Запрос и получение исходных данных, влияющих на адаптацию, от участвующих в проекте организаций |  |
| А.3. Задачи работы ***Выбор процессов, работ и задач*** | | |
| А.3.1 | Определение необходимых процессов, работ и задач с учетом разрабатываемой документации и обязанностей исполнителей |  |
| А.3.2 | Установка в договоре дополнительных процессов, работ и задач, не описанных в стандарте *СТБ ИСО/МЭК 12207-2003*, оценка соответствия организационных процессов |  |
| А.3.3 | Отдельный анализ на предмет исключения из ЖЦ обязательных (по стандарту *СТБ ИСО/МЭК 12207-2003*) задач, с учетом факторов риска, стоимости, графика работ, выполнимости, объема, критичности и интерфейса с пользователем |  |
| А.4. Задачи работы ***Документирование решений по адаптации и их обоснование*** | | |
| А.4.1 | Документальное оформление решений по адаптации с их обоснованием | Отчет |

К ***характеристикам условий выполнения проекта*** (см. задачу А.1.1) могут быть отнесены следующие:

* модель жизненного цикла;
* влияние жизненного цикла существующей системы;
* требования к системе и программным средствам;
* организационные подходы, процедуры и цели;
* размер, сложность, критичность и типы системы, программного продукта или программной услуги;
* методы разработки;
* количество задействованного персонала и участвующих в проекте сторон;
* методы и политика заказа.

*Приложение В* стандарта *СТБ ИСО/МЭК 12207-2003* содержит руководство по адаптации, поясняющее использование процесса адаптации к конкретному программному проекту.

В данном приложении рекомендуется выполнять *два уровня адаптации* *СТБ ИСО/МЭК 12207-2003*:

*первый уровень* – адаптация к конкретной области деятельности – медицинской, авиационной, военной и т.п.;

*второй уровень* – адаптация к каждому конкретному проекту или договору.

*Приложение В* может быть использовано для первого уровня адаптации. При этом рекомендуется:

* для встроенного в систему программного продукта определить, требуется ли от разработчика выполнение работ по созданию системы;
* для отдельно поставляемого продукта рассмотреть работы по созданию системы (см. работы 5.3.2, 5.3.3, 5.3.10, 5.3.11 на рис. 7), хотя они могут и не понадобиться.

*Приложение В* рекомендует также выполнять адаптацию оценок программных продуктов и работ, проводимых в течение жизненного цикла. Данные оценки группируются в ***пять категорий***:

1. *оценки внутри основных процессов*; выполняются персоналом, реализующим данные процессы; цель – текущая оценка полученных результатов;
2. *верификация и аттестация* (см. , 18); выполняется заказчиком, поставщиком или независимой стороной; цель – верификация и аттестация с различной степенью зависимости от проекта;
3. *совместные анализы и аудиторские проверки* (см. рис. 19, 20); выполняются проверяющей и проверяемой сторонами; цель – оценка состояния и соответствия продуктов и работ утвержденному графику;
4. *обеспечение качества* (см. рис. 16); выполняется персоналом, не зависящим от разработчиков программного продукта; цель – представление независимой гарантии соответствия программных продуктов и процессов требованиям договора и утвержденным планам;
5. *усовершенствование* (см. рис. 25); выполняется организацией без учета требований конкретного проекта или договора; цель – эффективное управление реализуемыми процессами и их усовершенствование.

Четыре первых категории оценок выполняются на проектном уровне, пятая категория – на организационном уровне. Данные категории оценок следует выбирать и адаптировать в соответствии с областью действия, величиной, сложностью и критичностью проекта, а также с учетом подходов к управлению процессами, существующих в организации.

Очевидно, что вопросы адаптации и применения *СТБ ИСО/МЭК 12207-2003* тесно связаны с ***основными характеристиками проекта***. К ним относятся:

1. *организационные подходы*; например, к машинным языкам, безопасности, защите, требованиям по резервированию технических средств и управлению риском;
2. *политика заказа*; например, типы договора, наличие подрядчиков, привлечение субподрядчиков и посредников по верификации и аттестации;
3. *концепция поддержки*; например, ожидаемая длительность поддержки, степень изменения продукта, сторона, выполняющая поддержку;
4. *модель жизненного цикла*; например, каскадная, эволюционная, инкрементная;
5. *вовлеченные стороны*; например, заказчик, поставщик, разработчик, субподрядчик, посредники по верификации и аттестации, персонал сопровождения; численность сторон; для больших проектов (десятки и сотни лиц) требуется соответствующий административный надзор и контроль, оценки, анализы, аудиторские проверки; для малых проектов такой контроль может не понадобиться;
6. *работы жизненного цикла системы*; например, подготовка проекта заказчиком, разработка и сопровождение поставщиком;
7. *характеристики системного уровня*; например, количество подсистем и объектов конфигурации;
8. *характеристики программного уровня*; например, количество программных объектов, типы, объемы и критичность программных продуктов, технические риски; выделяются следующие типы программных продуктов:

* новая разработка; должны учитываться все требования к процессу разработки;
* использование готового программного продукта; должна быть выполнена оценка функциональных характеристик, документации, применимости, возможность поддержки; процесс разработки может не понадобиться;
* модификация готового программного продукта; должна быть выполнена оценка функциональных характеристик, документации, применимости, возможность поддержки; процесс разработки реализуется с учетом критичности продукта и величины изменений;
* программный или программно-аппаратный продукт, встроенный или подключенный к системе; необходимо учитывать работы процесса разработки, связанные с системой;
* отдельно поставляемый программный продукт; не требуется учитывать работы процесса разработки, связанные с системой;
* непоставляемый программный продукт; требования стандарта *СТБ ИСО/МЭК 12207-2003* можно не учитывать;

1. *другие характеристики*; например, усиленный административный контроль за критичными или большими программными продуктами.

### 2.6. Инструментальные средства автоматизации жизненного цикла программных средств и систем

Очевидно, что большие размеры и высокая сложность разрабатываемых программных средств при ограничениях на бюджетные и временные затраты проекта могут привести к низкому качеству конечных программных продуктов и системы в целом. В этой связи в последнее время все большее внимание уделяется современным технологиям и инструментальным средствам, обеспечивающим автоматизацию процессов жизненного цикла программных средств (CASE-средствам). Использование таких инструментальных средств позволяет существенно сократить длительность и стоимость разработки систем и ПС при одновременном повышении качества процесса разработки и, как следствие, качества разработанных программных средств.

К современным инструментальным средствам, обеспечивающим эффективную поддержку процессов жизненного цикла программных средств и систем, можно отнести линейку интегрированных продуктов Telelogic. Основными в данной линейке являются следующие инструментальные средства.

2.6.1. Telelogic DOORS

Инструментальное средство ***Telelogic DOORS*** предназначено для автоматизации управления требованиями в жизненном цикле программных средств.

*Telelogic DOORS* может быть использовано при выполнении процессов заказа, поставки и разработки. Табл. 31 содержит работы и задачи вышеназванных процессов, которые может поддерживать *Telelogic DOORS*.

В данной таблице номера работ и задач ЖЦ ПС приведены в соответствии с их иерархическими номерами в стандарте *ISO/IEC 12207:1995* и его аутентичных аналогах *СТБ ИСО/МЭК 12207–2003* и *ГОСТ Р ИСО/МЭК 12207–99*.

В последнем столбце таблицы приведены номера таблиц пособия, содержащих описание соответствующих работ процессов жизненного цикла программных средств.

Таблица 31

**Работы и задачи процессов заказа, поставки и разработки,   
поддерживаемые инструментальным средством *Telelogic DOORS***

|  |  |  |  |
| --- | --- | --- | --- |
| Название процесса | Номер и  название работы | Номер и содержание  поддерживаемой задачи | Номер  таблицы  пособия |
| Заказ | 5.1.1. Подготовка процесса заказа | 5.1.1.1. Описание концепции или потребности в заказе  5.1.1.2. Анализ требований к системе  5.1.1.4. Анализ требований к программным средствам | 1 |
| Поставка | 5.2.4. Планирование | 5.2.4.1. Анализ требований к заказу | 2 |
| Разработка | 5.3.2. Анализ требований к системе | 5.3.2.1. Анализ области применения системы и определение требований к ней | 4 |
| 5.3.3. Проектирование системной архитектуры | 5.3.3.1. Распределение требований к системе между объектами технических и программных средств архитектуры и ручными операциями и дальнейшее уточнение требований | 5 |
| 5.3.4. Анализ требований к программным средствам | 5.3.4.1. Определение требований к программным средствам | 6 |
| 5.3.5. Проектирование программной архитектуры | 5.3.5.1. Распределение требований к программному объекту между его компонентами и дальнейшее уточнение требований | 7 |
| 5.3.6. Техническое проектирование программных средств | 5.3.6.1. Распределение технических требований к компонентам между программными модулями и дальнейшее уточнение требований | 8 |

Вышеприведенные задачи (табл. 31) непосредственно связаны с разработкой требований к системе, программным средствам или их элементам и могут быть автоматизированы с помощью *Telelogic DOORS*. Это позволяет повысить качество выполнения данных задач. Кроме того, следует иметь в виду, что для большинства других задач процессов жизненного цикла требования являются исходными данными. Поэтому структуризация требований и автоматизация их разработки также повышает качество выполнения таких задач, а следовательно, и всего процесса разработки систем и программных средств в целом.

2.6.2. Telelogic TAU

Семейство инструментальных средств ***Telelogic TAU*** предназначено для автоматизации работ и задач процесса разработки программных средств и систем. Данное семействопредоставляет визуальную среду разработки, основанную на использовании универсального языка моделирования UML, с возможностью последующей автоматической кодогенерации ПС. Использование семейства *Telelogic TAU* позволяет унифицировать язык общения между системными аналитиками, проектировщиками, программистами и другими разработчиками, ускорить и повысить качество проектирования ПС и систем, автоматизировать тестирование (испытания) программных модулей, компонентов, ПС и систем.

Семейство *Telelogic TAU* включает следующие инструментальные средства:

* ***TAU/Model Author*** – инструментальное средство для рисования диаграмм моделей;
* ***TAU/Architect*** – инструментальное средство для системных инженеров;
* ***TAU/Developer*** – инструментальное средство для разработчиков программных средств;
* ***TAU/Tester*** – инструментальное средство для тестировщиков.

Табл. 32 содержит работы и задачи процесса разработки, которые поддерживает семейство *Telelogic TAU*.

Таблица 32

**Работы и задачи процесса разработки,   
поддерживаемые семейством инструментальных средств *Telelogic TAU***

|  |  |  |
| --- | --- | --- |
| Номер и название работы | Номер и содержание поддерживаемой задачи | Номер  таблицы  пособия |
| 5.3.3. Проектирование системной архитектуры | 5.3.3.1. Определение общей архитектуры системы | 5 |
| 5.3.5. Проектирование программной архитектуры | 5.3.5.1. Преобразование требований к программному объекту в его архитектуру (разработка эскизного проекта)  5.3.5.2. Разработка эскизного проекта интерфейсов программного объекта и его компонентов  5.3.5.3. Разработка эскизного проекта базы данных | 7 |
| 5.3.6. Техническое проектирование программных средств | 5.3.6.1. Разработка технического проекта для компонентов программного объекта (представление их в виде набора программных модулей)  5.3.6.2. Разработка технического проекта интерфейсов программного объекта, его компонентов и модулей  5.3.6.3. Разработка технического проекта базы данных | 8 |
| 5.3.7. Программирование и тестирование программных средств | 5.3.7.1. Программирование программных модулей, базы данных  5.3.7.2. Тестирование программных модулей и базы данных | 9 |
| 5.3.8. Сборка программных средств | 5.3.8.2. Сборка программных модулей и компонентов и тестирование результатов сборки | 10 |
| 5.3.9. Квалификационные испытания программных средств | 5.3.9.1. Проведение квалификационных испытаний (тестирования) программных средств | 11 |
| 5.3.11. Квалификационные испытания системы | 5.3.11.1. Проведение квалификационных испытаний | 13 |

2.6.3. Telelogic SYNERGY

Инструментальное средство ***Telelogic SYNERGY*** предназначено для автоматизации управления изменениями и конфигурацией программных средств. Его использование позволяет упростить организацию коллективной работы над проектом и обеспечить контроль текущего состояния проекта с учетом возможных изменений в требованиях заказчика, в функциях разрабатываемых программных продуктов, исправлений ошибок в продуктах.

*Telelogic SYNERGY* обеспечивает поддержку изменений, вносимых в выходные результаты работ процессов разработки или сопровождения, автоматизирует процесс управления конфигурацией жизненного цикла ПС и систем. *Telelogic SYNERGY* также может быть использован при выполнении ряда задач других процессов ЖЦ (табл. 33).

Таблица 33

**Работы и задачи процессов жизненного цикла программных средств,   
поддерживаемые инструментальным средством *Telelogic SYNERGY***

|  |  |  |  |
| --- | --- | --- | --- |
| Название процесса | Номер и название работы | Номер и содержание поддерживаемой задачи | Номер таблицы пособия |
| Заказ | 5.1.3. Подготовка и корректировка договора | 5.1.3.5. Контроль изменений, вносимых в договор в ходе его реализации | 1 |
| 5.1.5. Приемка и закрытие договора | 5.1.5.3. Управление конфигурацией поставленного ПП |
| Поставка | 5.2.5. Выполнение и контроль | 5.2.5.3. Надзор за реализацией проекта, выявление и решение проблем | 2 |
| Документирование | 6.1.3. Выпуск | 6.1.3.2. Управление документированием в соответствии с процессом управления конфигурацией | 18 |
| 6.1.4. Сопровождение | 6.1.4.1. Внесение изменений в документацию согласно процессам сопровождения и управления конфигурацией | 18 |
| Создание  инфраструктуры | 7.2.3. Сопровождение инфраструктуры | 7.2.3.1. Сопровождение, контроль и изменение инфраструктуры для удовлетворения требований к процессу, обеспечение связи с процессом управления конфигурацией | 27 |

2.6.4. Telelogic DocExpress

Инструментальное средство ***Telelogic DocExpress*** предназначено для поддержки процесса документирования жизненного цикла программных средств. *Telelogic DocExpress* позволяет автоматизировать поступление из различных источников исходных данных для документирования, поддерживать исходные данные и документацию в актуальном состоянии, выполнять автоматическое форматирование документов в соответствии с действующими стандартами на документацию, упрощает сопровождение документов.

*Telelogic DocExpress* может использоваться при выполнении задач жизненного цикла, в которых предполагается документальное оформление выходных результатов (табл. 1 – 30 пособия, правый столбец).

2.6.5. Telelogic TAU Logiscope

Семейство инструментальных средств ***Telelogic TAU Logiscope*** входит в состав семейства *Telelogic TAU* и может использоватьсяинженерами по качеству и тестировщиками.

Семейство *TAU Logiscope* предназначено для поддержки задач жизненного цикла, связанных с выполнением проверок, оценок и тестирования промежуточных продуктов разработки. Применение *Telelogic TAU Logiscope* позволяет управлять качеством процессов, промежуточных и конечных продуктов жизненного цикла программных средств, сократить стоимость и сроки выполнения проектов.

*Telelogic TAU Logiscope* дает возможность автоматизировать поиск ошибок в исходных кодах программных модулей, компонент, объектов программных средств, оценить качество, сложность и соответствие принятым правилам исходных кодов, оценить степень покрытия исходных кодов тестами.

Семейство *Telelogic TAU Logiscope* включает следующие инструментальные средства:

* ***Logiscope Audit*** – инструментальное средство для оценки качества и графического анализа исходных программных кодов;
* ***Logiscope RuleChecker*** – инструментальное средство для проверки исходного кода на соответствие принятым правилам;
* ***Logiscope TestChecker*** – инструментальное средство для проверки степени покрытия исходного кода тестовыми наборами;
* ***Logiscope Reviewer*** – инструментальное средство, сочетающее в себе возможности *Logiscope Audit* и *Logiscope RuleChecker*.

Семейство *Telelogic TAU Logiscope* поддерживает вспомогательные процессы жизненного цикла, связанные с управлением качеством процессов и продуктов: обеспечения качества, верификации, аттестации, совместного анализа, аудита (см. рис. 3).

Табл. 34 содержит работы и задачи процессов поставки и разработки, при выполнении которых может использоваться семейство *Telelogic TAU Logiscope*.

Семейство *Telelogic TAU Logiscope* используется при выполнении работ и задач процесса разработки, связанных с исходными текстами программных модулей. Это касается, в первую очередь, работы «Программирование и тестирование программных средств» и, возможно, начальных этапов работы «Сборка программных средств», когда программные модули собираются в небольшие группы и выполняется тестирование этих групп.

В заключение следует отметить, что инструментальные средства семейства компании ***Telelogic*** могут интегрироваться друг с другом. Таким образом, их совместное использование поддерживает подавляющее большинство работ и задач процессов жизненного цикла программных средств.

Таблица 34

Работы и задачи процессов поставки и разработки,   
поддерживаемые семейством инструментальных средств Telelogic TAU Logiscope

|  |  |  |  |
| --- | --- | --- | --- |
| Название  процесса | Номер и название работы | Номер и содержание  поддерживаемой задачи | Номер таблицы пособия |
| Поставка | 5.2.6. Проверка и оценка | 5.2.6.1. Участие в совещаниях, подготовке приемки ПП, приемочных испытаниях, совместных анализах и аудиторских проверках  5.2.6.2. Выполнение верификации и аттестации ПП, программной услуги и процессов  5.2.6.3. Выполнение работ по обеспечению качества | 2 |
| Разработка | 5.3.7. Программирование и тестирование программных средств | 5.3.7.1. Программирование программных модулей, базы данных, разработка процедур и данных для их испытаний (тестирования)  5.3.7.4. Уточнение общих требований к тестированию и программы сборки программного средства  5.3.7.5. Оценка результатов программирования и тестирования по критериям | 9 |
| 5.3.8. Сборка программных средств | 5.3.8.5. Оценка планов и результатов сборки и тестирования по критериям | 10 |

## 3. Качество программных средств. 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 прогнозирования, оценки и управления 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.

В данной части пособия используются следующие ***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.

*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);
* *относительная* – соответствует упорядоченной шкале с равноудаленными делениями, оцененными в относительных единицах относительно некоторой абсолютной величины (обычно в диапазоне от 0 до 1).

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.

## 4. Стандартизация качества программного обеспечения в Республике Беларусь

### 4.1. Общие сведения о стандартах в области оценки качества, действующих на территории Республики 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*.** Данный стандарт под обозначением ***ГОСТ Р ИСО/МЭК 9126–93*** действует на территории России с 1994 г.

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 *СТБ ИСО/МЭК 12207-2003*, рассмотренными в разд. 2.

### 4.2. Классификация методов определения показателей качества программного средства

Стандарт *ГОСТ 28195–99* и его предыдущая версия *ГОСТ 28195–89* классифицируют 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.

### 4.3. 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 *четырехуровневую* 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,* частичноне соответствуют иерархической модели качества, принятой в международных стандартах.

В стандартах *ГОСТ 28806–90* и *СТБ ИСО/МЭК 9126–2003* определеныследующие ***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.

### 4.4. Метод оценки качества программных средств по ГОСТ 28195–99

Стандартом *ГОСТ 28195–99* и его предыдущей версией *ГОСТ 28195–89* рекомендован метод интегральной оценки качества программных средств, основанный на иерархической модели качества.

В соответствии с данным 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.

1. Процесс применения:

* фаза внедрения;
* фаза эксплуатации;
* фаза сопровождения.

´

Вышеприведенные фазы представляют собой временные периоды, соответствующие работам, совокупностям работ или процессам ЖЦ ПС, определенным стандартом *СТБ ИСО/МЭК 12207-2003* и описанным в разд. 2.

Оценка качества ПС заключается в выборе номенклатуры показателей, их оценке и сопоставлении с базовыми значениями.

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

1. факторы качества (в терминологии, принятой в международных стандартах, соответствуют характеристикам качества);
2. критерии качества (в международной терминологии – подхарактеристики качества);
3. метрики (соответствует международной терминологии);
4. оценочные элементы или единичные показатели (данный уровень в международных стандартах отсутствует).

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

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

В качестве примера на рис. 28 – приведены три верхних уровня иерархической модели фактора *Сопровождаемость* для различных фаз жизненного цикла. Номера на данных рисунках соответствуют номерам метрик данного фактора.

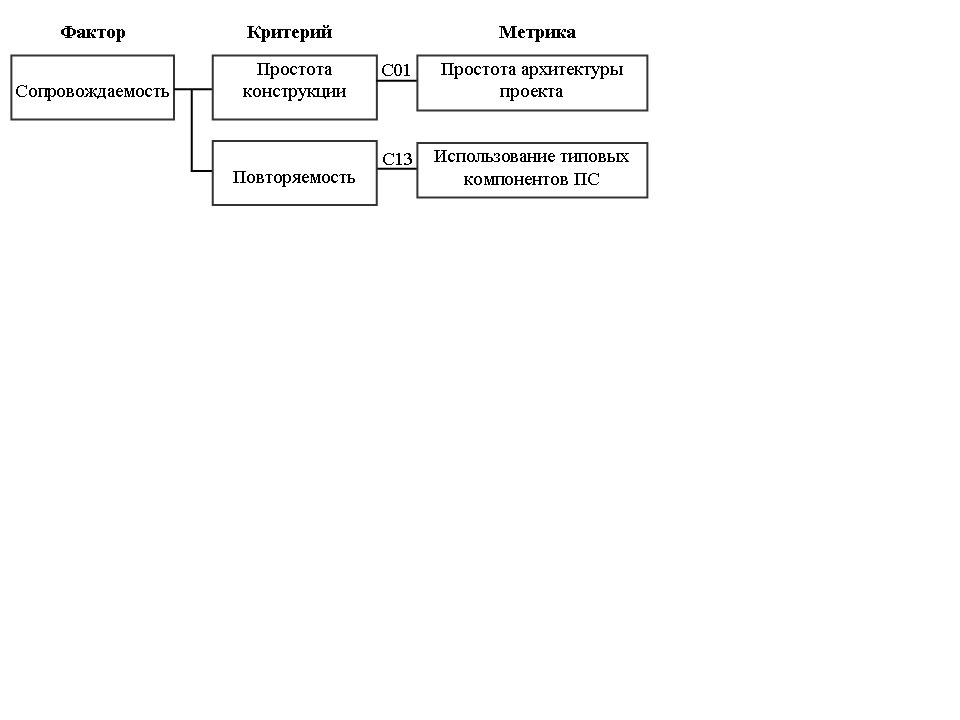


Рис. . Модель сопровождаемости для фазы анализа



Рис. . Модель сопровождаемости для фазы проектирования

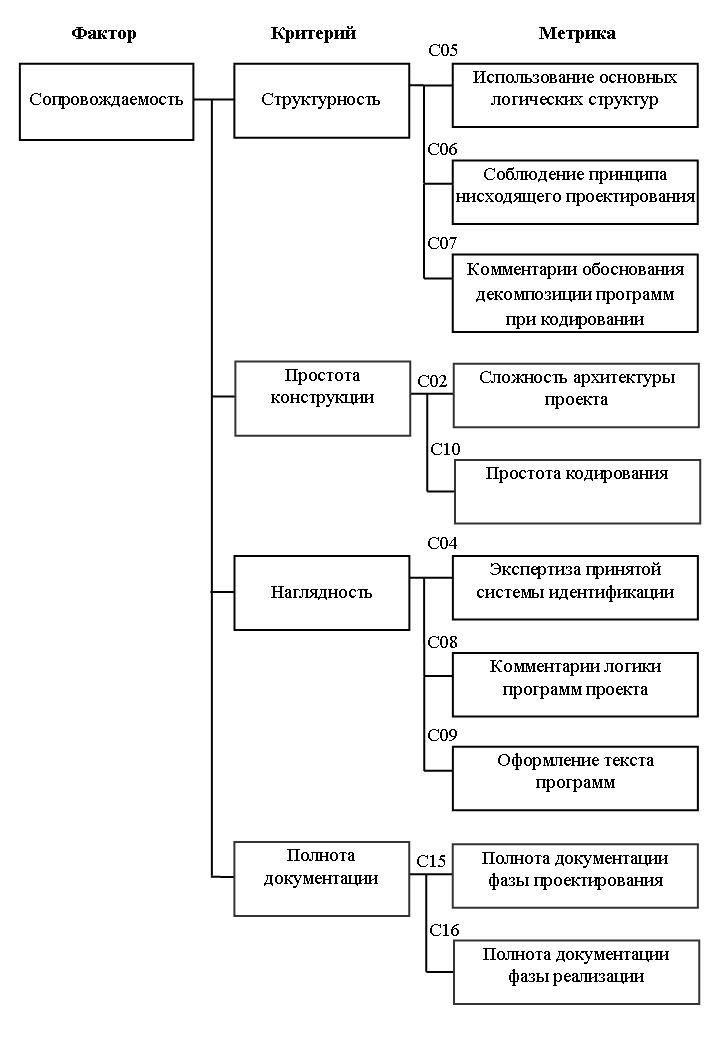


Рис. . Модель сопровождаемости для фазы реализации

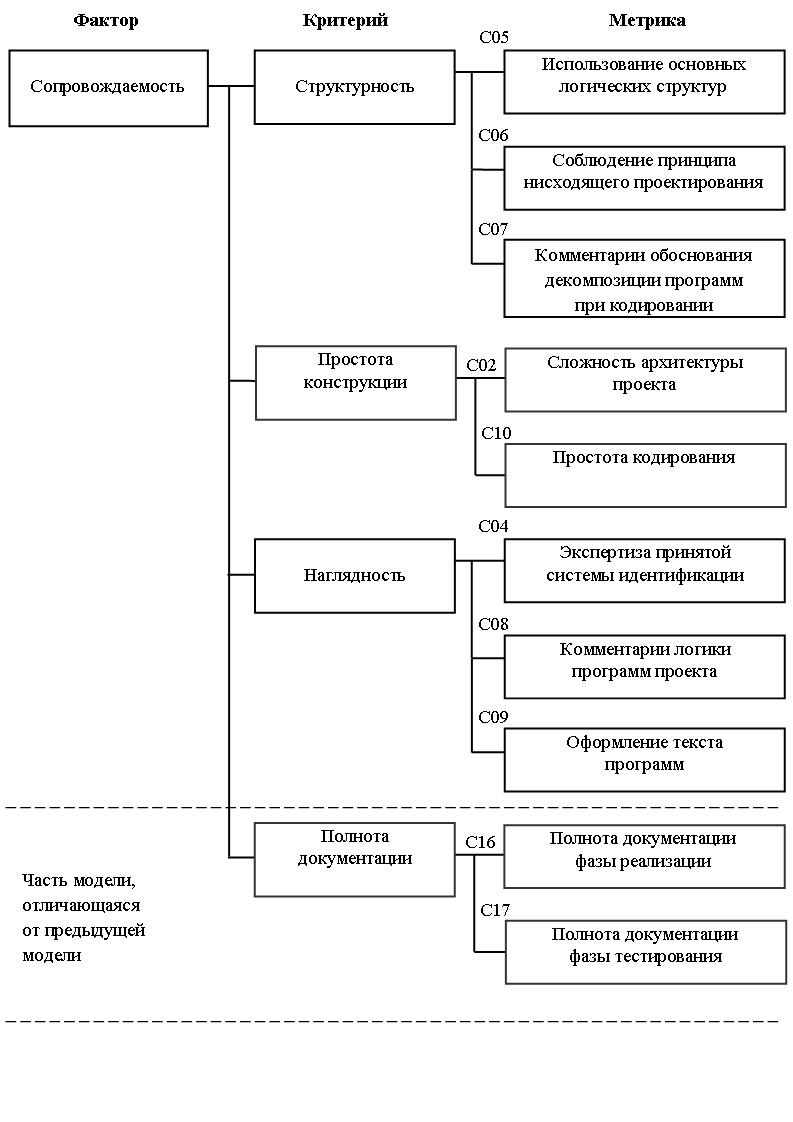


Рис. . Модель сопровождаемости для фазы тестирования

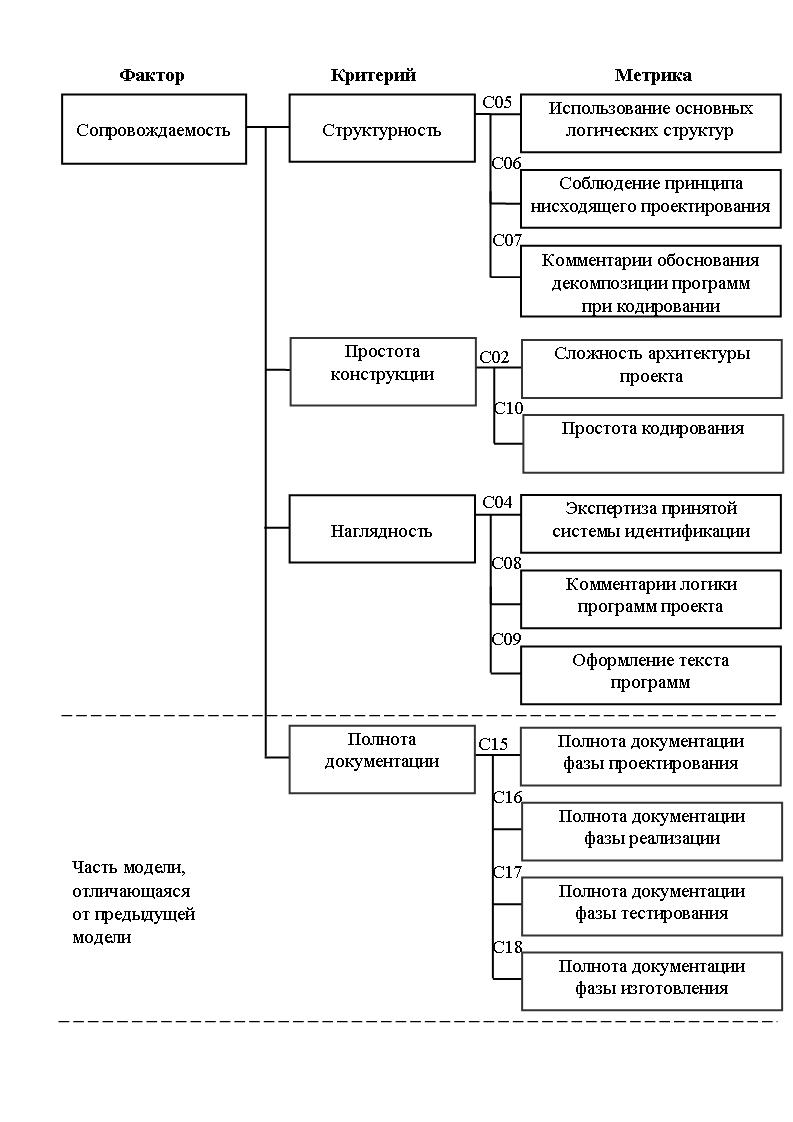


Рис. . Модель сопровождаемости для фазы изготовления

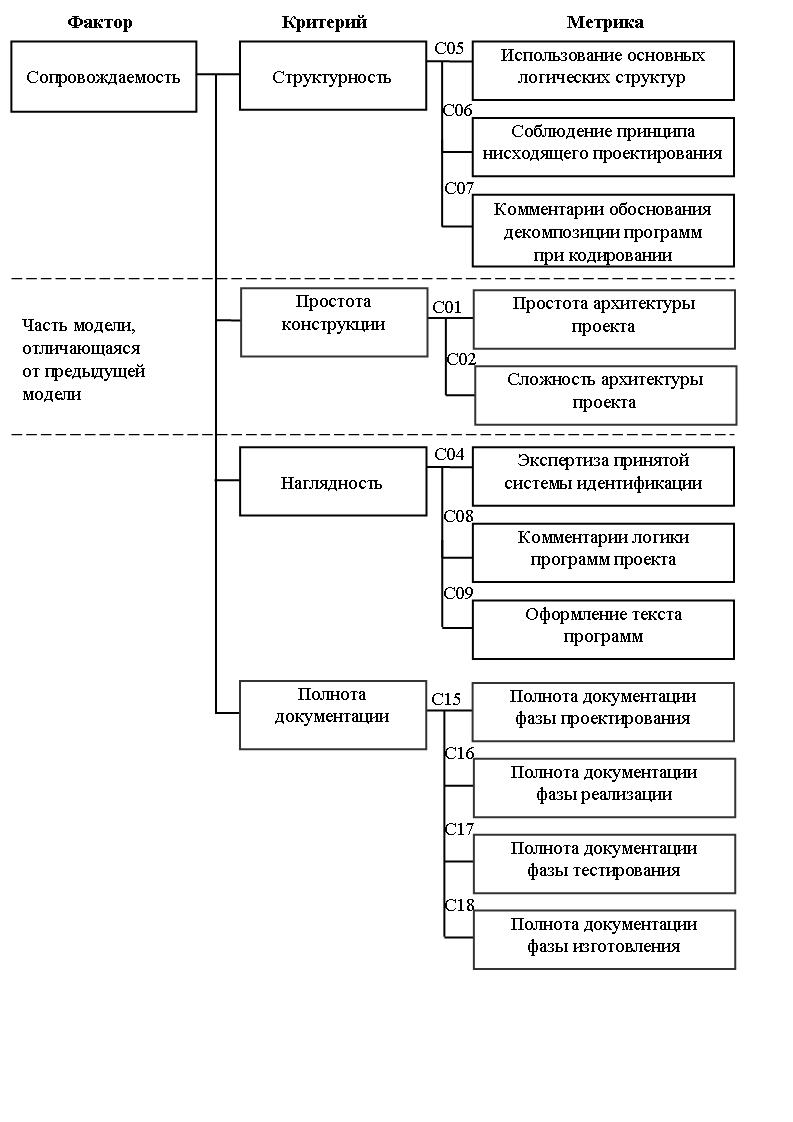


Рис. . Модель сопровождаемости для фазы сопровождения

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.

Табл. 35 содержит перечень оценочных элементов для фактора *Сопровождаемость*. 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 *Сопровождаемости* номера метрик обозначены на рис. 29 – ). Четвертый и пятый символы – порядковый номер данного оценочного элемента в метрике.

Таблица 35

**Оценочные элементы фактора *Сопровождаемость***

| Код элемента | Наименование | | Метод оценки |
| --- | --- | --- | --- |
| С0101 | Наличие модульной схемы программы | | Экспертный |
| С0102 | Оценка программы по числу уникальных модулей | | Экспертный |
| С0201 | Наличие ограничений на размеры модуля | | Экспертный |
| С0301 | Наличие проверки корректности передаваемых данных | | Экспертный |
| С0302 | Оценка простоты программы по числу точек входа и выхода ***W:***  Форм16  где ***D*** – общее число точек входа в программу; ***F*** – общее число точек выхода из программы | | Измерительный + расчетный |
| С0303 | Осуществляется ли передача  результатов работы модуля через вызывающий его модуль | | Экспертный |
| С0304 | Осуществляется ли контроль за правильностью данных, поступающих в вызывающий модуль от вызываемого | | Экспертный |
| С0305 | Наличие требований к независимости модулей программы от типов и форматов выходных данных | | Экспертный |
| С0401 | Наличие требований к системе идентификации | | Экспертный |
| С0501 | Наличие требований по использованию основных логических структур | | Экспертный |
| С0601 | Использование при построении программ метода структурного программирования | | Экспертный + измерительный |
| С0602 | Соблюдение принципа разработки программы сверху вниз | | Экспертный |
| С0603 | Оценка программы по числу циклов с одним входом и одним выходом | | Экспертный + измерительный |
| С0604 | Оценка программы по числу циклов | | Экспертный + измерительный |
| С0701 | Наличие комментариев обоснования декомпозиции программ при кодировании | | Экспертный |
| С0801 | Наличие комментариев ко всем машинозависимым частям программы | | Экспертный |
| С0802 | Наличие комментариев ко всем машинозависимым операторам программы | | Экспертный |
| С0803 | Наличие комментариев в точках входа и выхода программы | | Экспертный |
| С0901 | Соответствие комментариев принятым соглашениям | | Экспертный |
| С0902 | Наличие комментариев-заголовков программы с указанием ее структурных и функциональных характеристик | | Экспертный |
| С0903 | Оценка ясности и точности описания последовательности функционирования всех элементов программы | | Экспертный |
| С1001 | | Используется ли язык высокого уровня | Экспертный |
| С1002 | | Оценка простоты программы по числу переходов по условию:  U=(1 – A/B),  где ***А*** – общее число переходов по условию; ***В*** – общее число исполняемых операторов | Измерительный + расчетный |
| С1301 | | Использование типовых компонентов ПС | Экспертный |
| С1401 | | Использование типовых проектных решений | Экспертный |
| С1501 | | Наличие программных спецификаций и требований, предъявляемых к программным средствам | Экспертный |
| С1502 | | Наличие документов, содержащих детальное описание принятых проектных решений | Экспертный |
| С1503 | | Наличие заключений по принятым проектным решениям, требованиям и спецификациям | Экспертный |
| С1601 | | Наличие описания и схемы иерархии модулей программы | Экспертный |
| С1602 | | Наличие описания основных функций | Экспертный |
| С1603 | | Наличие описания частных функций | Экспертный |
| С1604 | | Наличие описания данных | Экспертный |
| С1605 | | Наличие описания алгоритмов | Экспертный |
| С1606 | | Наличие описания интерфейсов | Экспертный |
| С1607 | | Наличие описания интерфейсов с пользователем | Экспертный |
| С1608 | | Наличие описания используемых числовых методов | Экспертный |
| С1609 | | Наличие описания всех параметров | Экспертный |
| С1610 | | Наличие описания методов настройки системы | Экспертный |
| С1611 | | Наличие описания всех диагностических сообщений | Экспертный |
| С1612 | | Реализация всех исходных модулей | Экспертный |
| С1701 | | Наличие описания всех диагностических сообщений эталонного образца | Экспертный |
| С1702 | | Наличие требований к тестированию программ | Экспертный |
| С1703 | | Достаточность требований к тестированию программ | Экспертный |
| С1801 | | Наличие описания процедуры изготовления эталонного образца | Экспертный |
| С1802 | | Наличие описания процедуры изготовления рабочих копий | Экспертный |
| С1803 | | Наличие описания процедуры контроля на идентичность рабочих копий с эталонным образцом | Экспертный |

*Оценка качества программного средства проводится в следующей последовательности:*

1. На фазе анализа проводится 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 **;**
       - относительного показателя качества **,**

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

Относительныйпоказатель качестваявляется функцией показателяи его базового значения ****и определяется по формуле

****.

*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 ПС должны соответствовать значениям показателей, отражающих современный уровень качества и прогнозируемый мировой уровень.

1. Каждый показатель качества второго и третьего уровней характеризуется двумя параметрами:
   * + - количественным значением;
       - весовым коэффициентом .

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

1. Определениеусредненной оценки оценочного элемента по нескольким его значениям (измерениям)  осуществляется по следующей формуле (*формула для вычисления значений показателей качества 4-го уровня*):

**,**

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

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

,

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

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

,

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

1. Относительные значения  ***j***-го критерия ***i***-го фактора по отношению к базовому значению определяются по формуле

.

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

,

,

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

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

*Достоинства* метода оценки качества, основанного на иерархической модели:

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 для тех показателей, которые такую оценку допускают.

### 4.5. 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 (см. подразд. 4.3). Второй уровень составляют подхарактеристики и третий – метрики качества.

Модель процесса оценки, положенная в основу рассматриваемого метода, приведена на рис. 35. Данная модель отражает основные стадии и этапы, требуемые для оценки качества ПС.

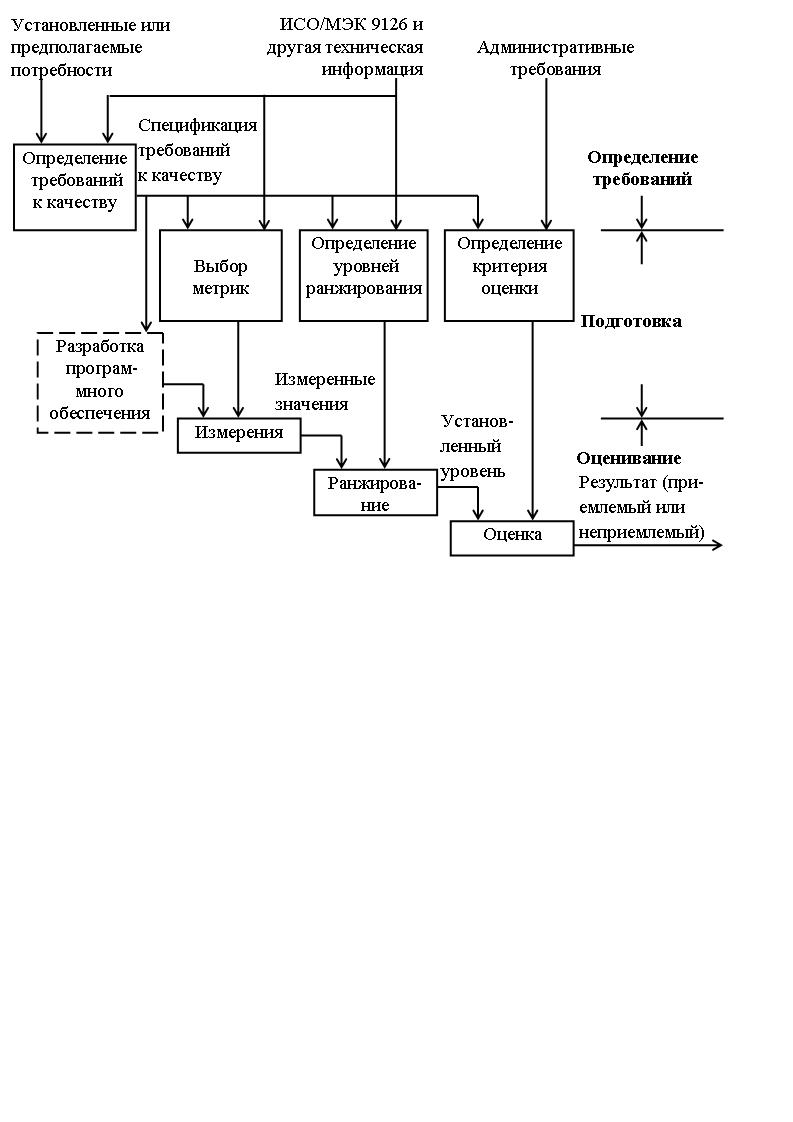


Рис. . Модель процесса оценки по СТБ ИСО/МЭК 9126–2003

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.

В стандарте *ISO/IEC 14598–1:1999* приведен пример следующих диапазонов ранжирования ():

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

Текущий уровень определяется для управления тем, чтобы новая система не становилась хуже по сравнению с существующей. Запланированный уровень определяет уровень, который считается достижимым при доступных ресурсах. Уровень худшего случая определяет границу принятия пользователем в случае, если изделие не удовлетворяет запланированному уровню (см. ). Так как качество ПС связано с конкретными потребностями, общие уровни ранжирования невозможны и должны определяться для каждого конкретного оценивания.

*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.

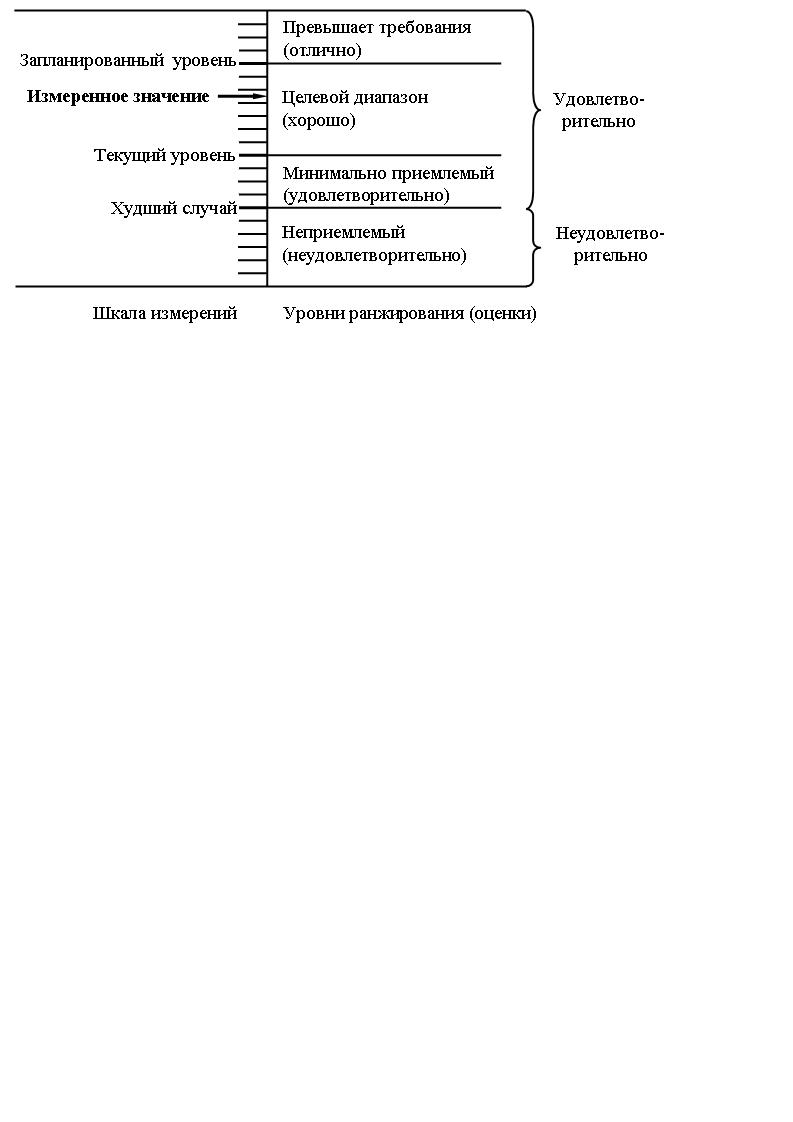


Рис. . Варианты ранжирования измеренных значений метрик   
по ISO/IEC 14598–1:1999

***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. ).

*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.

## 5. Стандартизация качества программных средств за рубежом

### 5.1. Стандарты серии 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***. В разд. 4 пособия описаны положения стандарта *СТБ ИСО/МЭК 9126–2003*, являющегося 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:***

* *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 стандарта* ***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 (см. подразд. 4.3). Однако в отличие от *ISO/IEC 9126:1991* подхарактеристики второго уровня стали нормативными, а не рекомендуемыми, определены две части модели качества (модель внутреннего и внешнего качества и модель качества в использовании) и исключен процесс оценки качества (он теперь содержится в стандарте *ISO/IEC 14598*). В данной части стандарта регламентированы следующие ***виды метрик***:

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

Вторая – четвертая части стандарта *ISO/IEC TR 9126–2–4* в настоящее время еще находятся в стадии разработки. Поэтому они 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 пояснения к применению метрик, к типам шкал метрик и типам измерений, примеры метрик для каждой подхарактеристики, примеры применения метрик на протяжении ЖЦ ПС.

*Во второй части стандарта* ***ISO/IEC TR 9126–2:2003*** определяются метрики количественного измерения 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.

*В третьей части стандарта* ***ISO/IEC TR 9126–3:2003*** определяются метрики количественного измерения внутреннего качества ПС. ***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.

*В четвертой части стандарта* ***ISO/IEC TR 9126–4:2004*** определяютсяметрики количественного измерения 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. 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. Данный стандарт состоит из шести частей:

* *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 стандарта* ***ISO/IEC 14598–1:1999*** приведен обзор остальных частей, определена связь *ISO/IEC 14598* со стандартами *ISO/IEC 9126–1–4* и *ISO/IEC 12207:1995.* В данной части содержатся общие требования к спецификации и оценке качества, разъясняются концепции оценки. Устанавливаются требования к методам измерений и оценки программных продуктов. Определяется общий процесс оценкикачества программного продукта (см. подразд. 5.5).

*Вторая часть стандарта* ***ISO/IEC 14598–2:2000*** содержит требования и руководство по поддержке оценки. В данной части приводятся концепции планирования и управления процессом оценки качества программного продукта, рассматривается содержание плана количественной оценки качества. Эта часть стандарта предназначена для применения на уровне организации или ее подразделений.

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

*Четвертая часть стандарта* ***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 ПП.Рассмотрены особенности выполнения оценки для имеющегося в наличии готового ПП, для приемки заказного ПП и для выбора из альтернативного числа готовых ПП.

*Пятая часть стандарта* ***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 приводятся концепции оценки и требования к процессу оценки. Рассмотрена структура отчета об оценке. Приводятся рекомендации по выбору уровней ранжирования при проведении измерений. Оценка качества программного продукта оценщиком может выполняться по запросу разработчика, заказчика (покупателя) или другой стороны.

*Шестая часть стандарта* ***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, внутренние или внешние метрики качества), методики и процедуры оценки, входные данные, связанные с оценкой, 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 может быть использована организациями, разрабатывающими новые модули оценки и производящими оценку ПС.

### 5.2. Связь качества программного средства с его жизненным циклом

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.

иллюстрирует изменение и взаимосвязь различных видов качества в жизненном цикле программных средств.

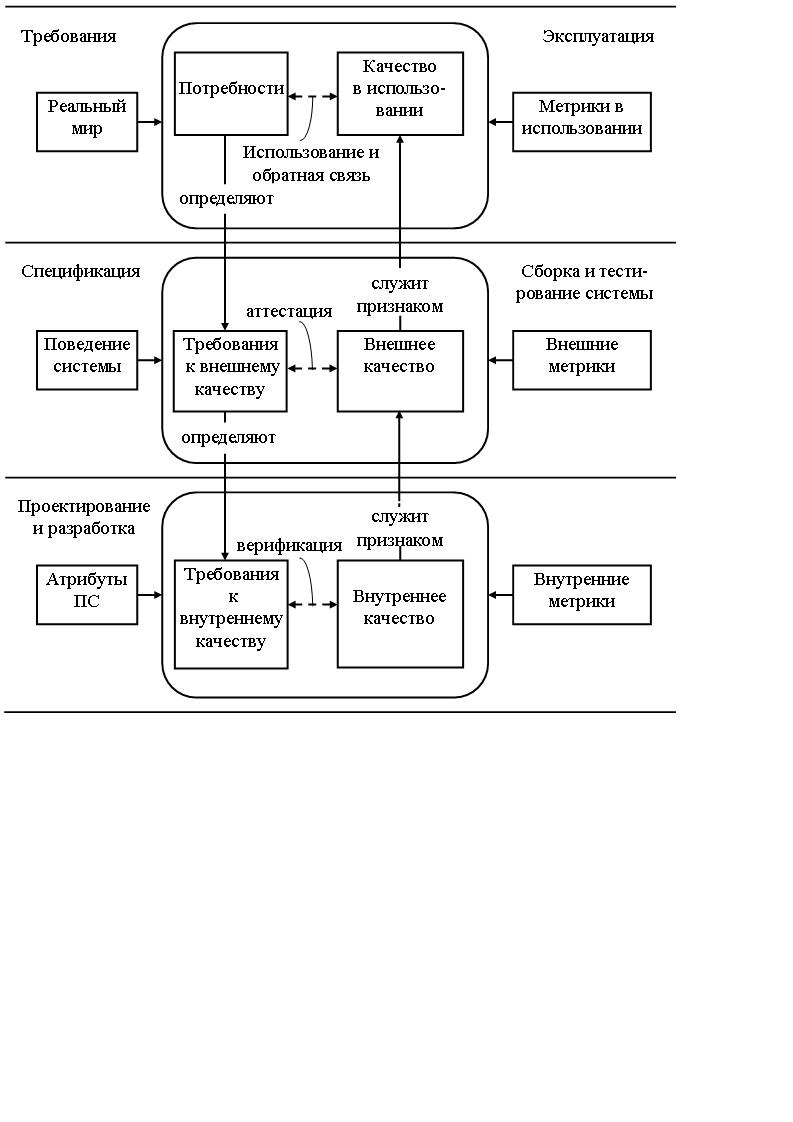


Рис. . Качество в жизненном цикле программных средств   
по ISO/IEC 14598–1:1999

### 5.3. Модель внешнего и внутреннего качества ПС

Как уже было отмечено, в стандарте *ISO/IEC 9126–1:2001* регламентированы *две части модели качества программных средств*: модель внутреннего и внешнего качества и модель качества в использовании. Данные модели различаются в зависимости от представления качества в ЖЦ ПС (см. ). 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 завершенного ПС.

На рис. 38 приведены два верхних уровня модели внешнего и внутреннего качества. На верхнем уровне находится ***шесть основных характеристик качества программных средств***. Это *функциональность, надежность, практичность, эффективность, сопровождаемость* и *мобильность*.

Подхарактеристики, находящиеся на втором уровне, также стандартизированы. Они могут измеряться внутренними или внешними метриками.

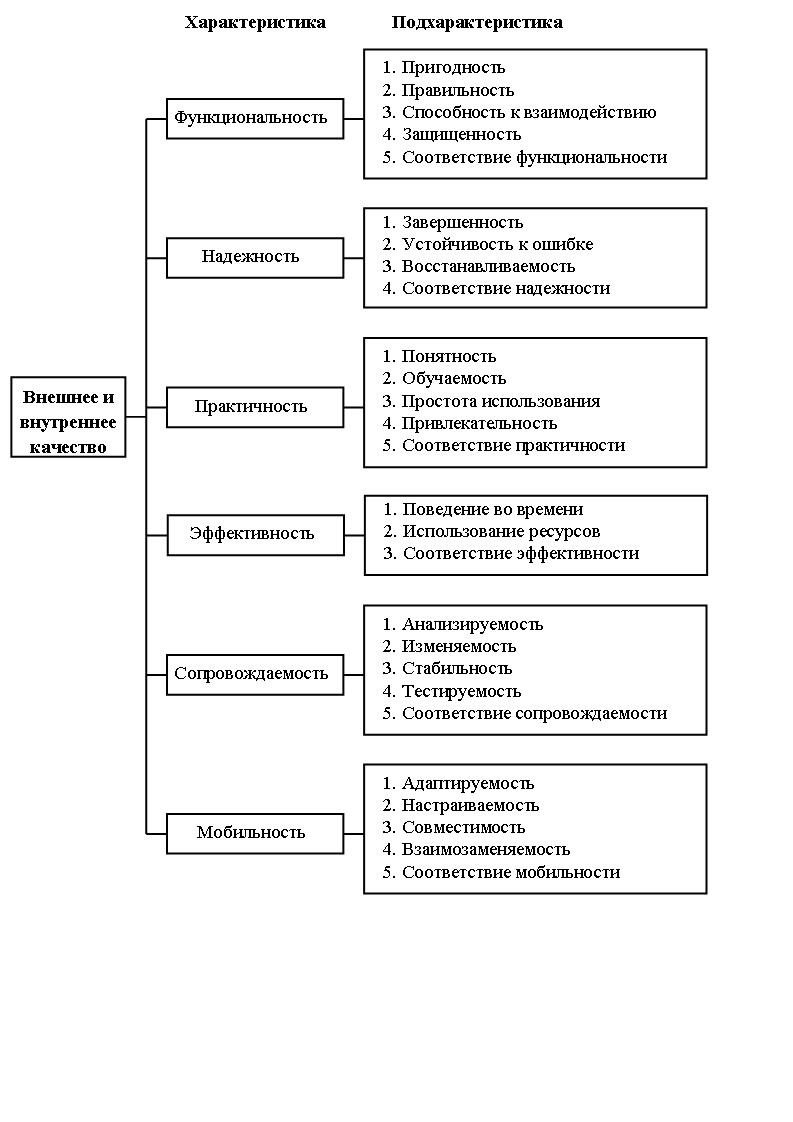


Рис. . Модель внешнего и внутреннего качества по ISO/IEC 9126–1:2001

5.3.1. Функциональность

**Функциональность** **(Functionality)** – способность программного продукта обеспечивать функции, удовлетворяющие установленные и подразумеваемые потребности при применении программного средства в заданных условиях. Эта характеристика определяет, *что* делает ПС в соответствии с потребностями. Другие характеристики определяют, *когда* и *как* эти потребности удовлетворяются. Подхарактеристики *Функциональности* приведены на рис. 38.

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

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

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

***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.

***Соответствие функциональности (Functionality compliance)*** – способность программного продукта соответствовать стандартам, соглашениям или нормам законов, связанным с функциональностью.

5.3.2. Надежность

**Надежность (Reliability)** – способность программного продукта поддерживать заданный уровень качества функционирования при его использовании в заданных условиях. Ограничения надежности в процессе эксплуатации вызваны ошибками в требованиях, проектировании и кодировании.

Подхарактеристики *Надежности* приведены на рис. 38.

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

***Устойчивость к ошибке (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. Одним из показателей восстанавливаемости является длительность восстановления.

***Готовность (доступность, Availability)*** – способность программного продукта быть в состоянии выполнять требуемую функцию в данный момент времени при заданных условиях использования. Готовность может оцениваться отношением времени, в течение которого ПС находится в работоспособном состоянии, к общему времени применения. Следовательно, готовность – это комбинация завершенности (которая влияет на частоту отказов), устойчивости к ошибке и восстанавливаемости (которая влияет на продолжительность восстановления). По этой причине готовность не включена в модель внутреннего и внешнего качества в виде отдельной подхарактеристики (см. рис. 38).

***Соответствие надежности (Reliability compliance)*** – свойство программного продукта соответствовать стандартам, соглашениям и нормативным документам, связанным с надежностью.

5.3.3. Практичность

**Практичность (Usability)** – способность программного продукта быть понятным, изученным, использованным и привлекательным для пользователя при применении в заданных условиях. Очевидно, что практичность зависит от некоторых аспектов таких характеристик качества, как функциональность, надежность и эффективность. Подхарактеристики *Практичности* приведены на рис. 38.

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

***Обучаемость (Learnability)*** – способность программного продукта, обеспечивающая изучение пользователем принципов его применения.

***Простота использования (Operability)*** – способность программного продукта, позволяющая пользователю эксплуатировать его и управлять им. На простоту использования влияют некоторые аспекты таких подхарактеристик, как пригодность, изменяемость, адаптируемость и простота внедрения. Простота использования зависит, например, от контролируемости и устойчивости к ошибке.

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

***Соответствие практичности (Usability compliance)*** – свойство программного продукта соответствовать стандартам, соглашениям и руководствам, связанным с практичностью.

5.3.4. 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. 38.

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

***Использование ресурсов (Resource utilisation)*** – способность программного продукта использовать соответствующее количество всех типов ресурсов при выполнении своих функций в заданных условиях. В данной подхарактеристике человеческие ресурсы не учитываются. Они учитываются в такой характеристике качества в использовании, как продуктивность (см. подразд. 5.4).

***Соответствие эффективности (Efficiency compliance)*** – способность программного продукта соответствовать стандартам и соглашениям, связанным с эффективностью.

5.3.5. Сопровождаемость

**Сопровождаемость (Maintainability)** – способность программного продукта к модификации. Модификации могут включать исправления, усовершенствования или адаптацию ПС к изменениям в среде применения, в требованиях и функциональных спецификациях. Подхарактеристики *Сопровождаемости* представлены на рис. 38.

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

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

***Стабильность (Stability)*** – способность программного средства предотвращать непредвиденные эффекты от его модификации.

***Тестируемость (Testability)*** – способность программного продукта к проверке результата модификации.

***Соответствие сопровождаемости (Maintainability compiance)*** – способность программного продукта соответствовать стандартам или соглашениям, связанным с сопровождением.

5.3.6. Мобильность

**Мобильность (Portability)** – способность программного продукта к переносу из одной среды в другую. Среда может включать организационное, аппаратное и программное окружение. Подхарактеристики *Мобильности* представлены на рис. 38.

***Адаптируемость (Adaptability)*** – способность программного продукта к адаптации 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.

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

***Взаимозаменяемость (Replaceability)*** – способность программного продукта к использованию вместо другого (заданного) ПП с той же целью и в той же среде. Например, для пользователя важна взаимозаменяемость новой версии ПП с его старой версией. Взаимозаменяемость может включать атрибуты таких подхарактеристик, как настраиваемость и адаптируемость.

***Соответствие мобильности (Portability compiance)*** – способность программного продукта соответствовать стандартам или соглашениям, связанным с мобильностью.

### 5.4. Модель качества в использовании

Второй частью модели качества, определенной в стандарте *ISO/IEC 9126–1:2001*, 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. ).

**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 ПП. Оно измеряется больше на основе результатов использования ПП в среде, чем на основе собственных внутренних свойств ПП.

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. 38) уровень подхарактеристик в модели качества в использовании отсутствует.

На рис. 39 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, позволяющая пользователям достигать заданные цели с точностью и полнотой в заданном контексте использования.

***Продуктивность (Productivity)*** – это способность программного продукта, позволяющая пользователям расходовать количество ресурсов, соответствующее результативности, достигаемой в заданном контексте использования. Ресурсы могут включать время выполнения задачи, усилия пользователя, материалы, стоимость использования.

***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 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 продукта.

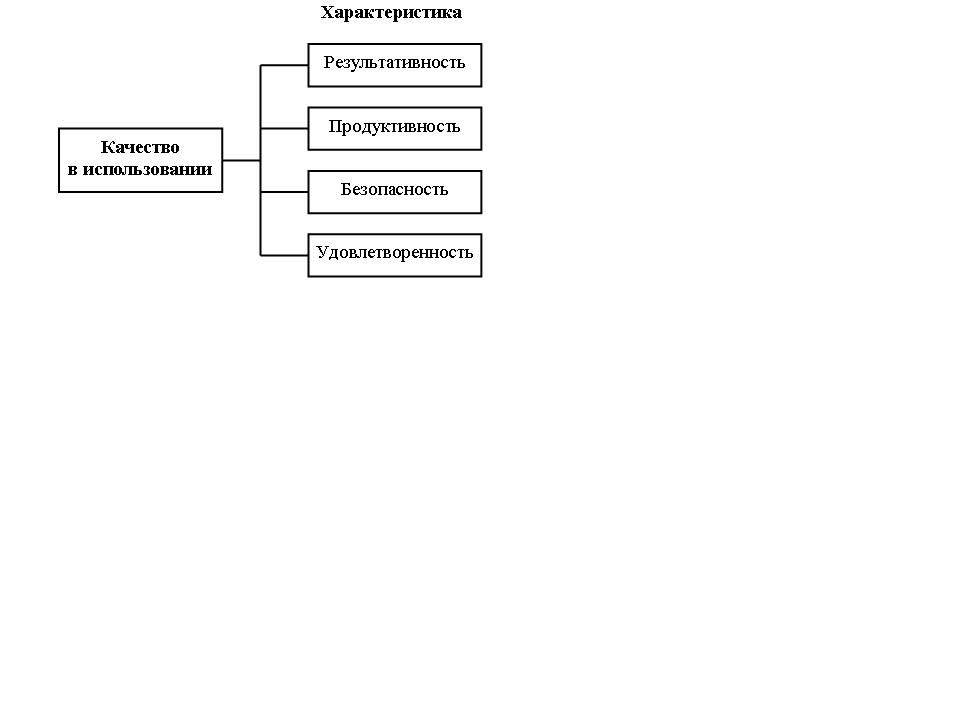
****

Рис. . Модель качества в использовании

### 5.5. 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. 40 приведена схема процесса оценки качества ПС. Процесс оценки состоит из *четырех 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.

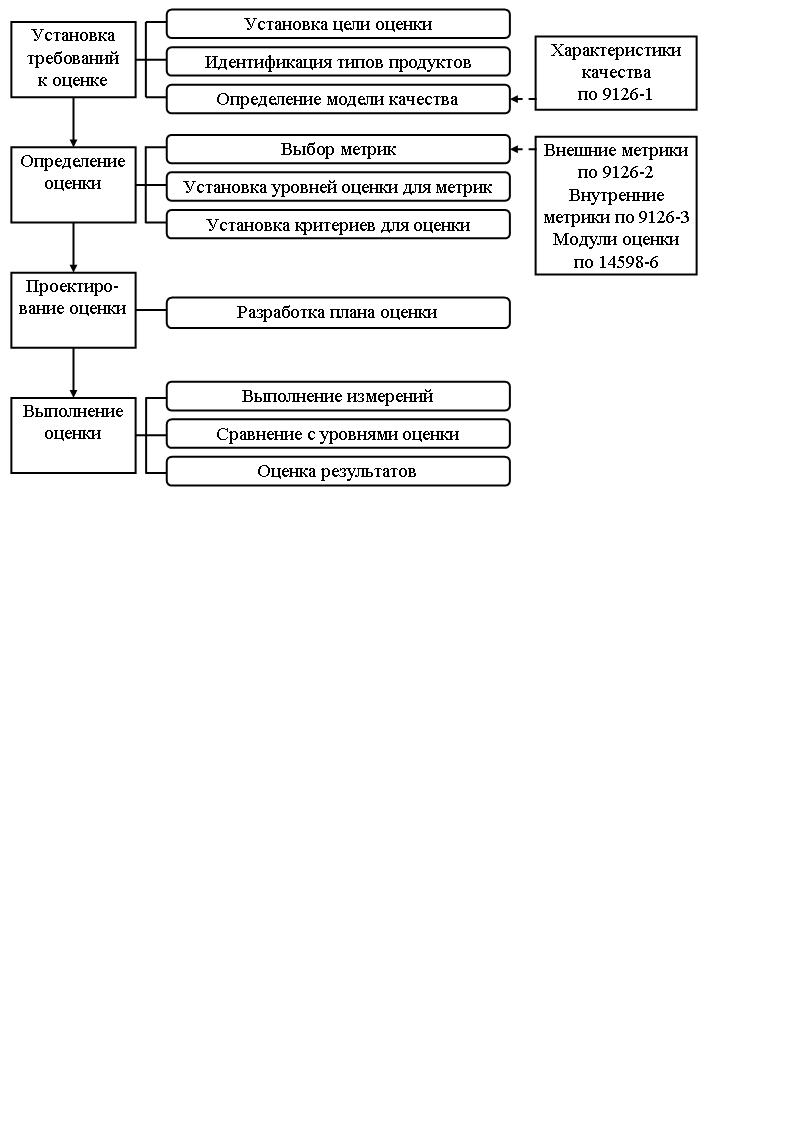


Рис. . Процесс оценки качества по ISO/IEC 14598–1:1999

*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 *СТБ ИСО/МЭК 12207-2003*.

*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 (см. подразд. пособия).

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 (см. ). 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* (см. рис. 38, 54 пособия). 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* (см. подразд. 6.2 – 6.4).

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. ).

*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/IEC14598–5:1998*. 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 (например, как показано на ).

*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.

## 6. Метрология качества программных средств

### 6.1. Свойства и критерии обоснованности метрик качества программных средств

Как было описано в предыдущем разделе, стандарт *ISO/IEC 9126–1:2001*классифицируетметрики качества ПС на внутренние, внешние и метрики качества в использовании. В модели внешнего и внутреннего качества метрики находятся на третьем уровне иерархии и определяют значения подхарактеристик качества. В модели качества в использовании метрики находятся на втором уровне иерархии и непосредственно определяют значения характеристик качества. Применение конкретного вида метрик определяется стадией жизненного цикла программного средства.

Вторая, третья и четвертая части стандарта *ISO/IEC TR 9126–2–4* посвящены детальному рассмотрению соответственно внешних и внутренних метрик качества программных средств и метрик 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:

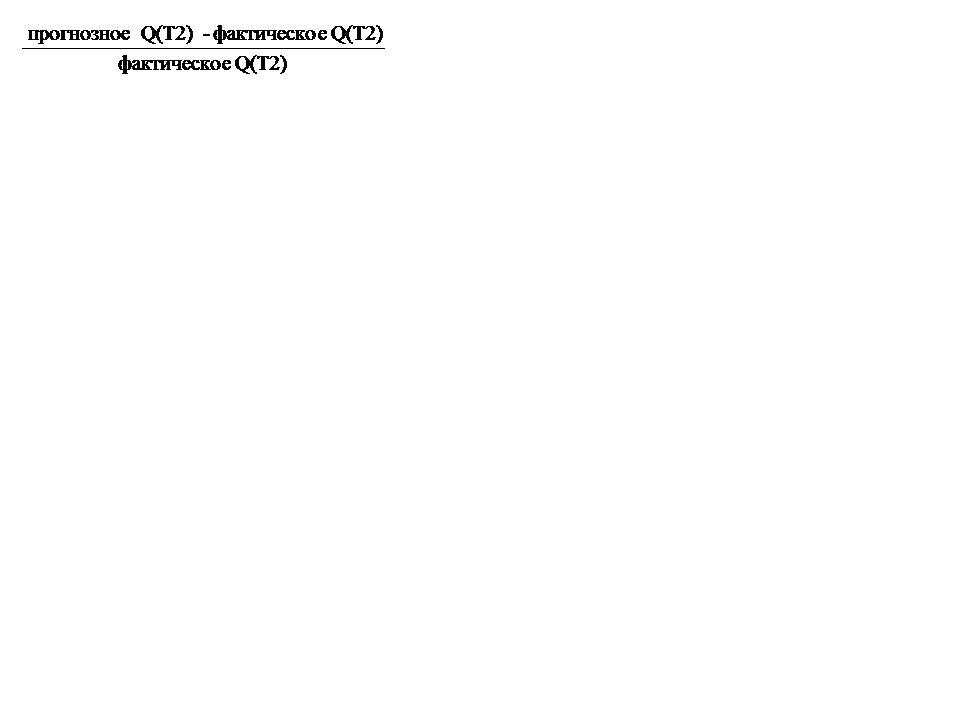
* *объективность*; результаты метрики и её входные данные должны быть основаны на фактах и не подвластны чувствам или мнениям специалистов по оценке или тестированию (исключая метрики удовлетворенности или привлекательности, с помощью которых измеряются чувства и мнения пользователя);
* *беспристрастность*; 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;

1. *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 (T1), имеющейся в момент времени T1, к величине Q (T2), полученной в момент времени Т2, должно сопровождаться изменением значения метрики от М (T1) до М (T2) 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> ... > Мn.
4. *предсказуемость*; 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



должна попадать в допустимый диапазон ошибок прогнозирования;

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

В стандартах *ISO/IEC 9126–2,–3,–4*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);

1. источники входных данных для измерения;
2. ссылка на *ISO/IEC 12207:1995* (процессы жизненного цикла программных средств, при выполнении которых применима метрика);
3. целевая аудитория (под аудиторией понимается категория пользователей, предъявляющих к некоторой документации одинаковые или аналогичные требования, определяющие содержание, структуру и назначение данной документации).

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* *по возможности* представляются в относительных единицах в виде

 (1)

или

, (2)

где ***Х*** – значение метрики; ***А*** – абсолютное (измеренное) значение некоторого свойства (атрибута) оцениваемого продукта или документации; ***В*** – 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 (1) или (2) позволяет привести их относительные значения в диапазон

, (3)

что упрощает их совместное использование при интегральной оценке качества программных средств.

В подразд. 6.2 – 6.4 приведены примеры метрик (по одной на каждую подхарактеристику или характеристику качества) из рекомендуемых в стандартах *ISO/IEC TR 9126–2,–3,–4.*

Следует отметить, что не все метрики, приведенные в стандартах *ISO/IEC TR 9126–2,–3,–4*, удовлетворяют вышеприведенным свойствам, критериям, оцениваются с помощью выражений (1), (2) или попадают в диапазон (3).

### 6.2. 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 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.

Табл. 36 содержит примеры внутренних метрик качества ПС, содержащихся в стандарте *ISO/IEC TR 9126–3:2003*. Во втором столбце таблицы по каждой подхарактеристике приведено название одной метрики, уникальная формула или номер формулы (1) или (2) из подразд. 6.1 для оценки данной метрики. Исходные данные в третьем столбце – это данные, используемые в соответствующей формуле для вычисления значения представленной метрики.

Таблица 36

**Внутренние метрики качества программных средств**

|  |  |  |
| --- | --- | --- |
| Название  подхарактеристики | Название метрики,  формула для ее оценки | Исходные данные для вычисления метрики  по соответствующей формуле |
| ***Функциональность*** | | |
| Пригодность | Полнота функциональной  реализации (2) | ***А*** – число нереализованных функций, обнаруженных при оценке; ***В*** – число функций, описанных в спецификации требований |
| Правильность | Точность  (1) | ***А*** – количество элементов данных, реализованных с заданными уровнями точности, подтвержденное при оценке; ***В*** – количество элементов данных, для которых в спецификации заданы уровни точности |
| Способность к взаимодействию | Соответствие интерфейсов (протоколов) (1) | ***А*** – количество интерфейсных протоколов, реализующих заданный в спецификации формат, подтвержденных при проверке; ***В*** – количество интерфейсных протоколов, которые должны быть реализованы в соответствии со спецификацией |
| Защищенность | Предотвращение  разрушения данных (1) | ***А*** – количество реализованных случаев предотвращения разрушения данных из заданных в спецификации, подтвержденное при проверке; ***В*** – количество случаев обработки (доступа), которые определены в спецификации как способные разрушить данные |
| Соответствие функциональности | Соответствие  функциональности (1) | ***А*** – количество корректно реализованных элементов, связанных с соответствием функциональности, подтвержденное при оценке; ***В*** – общее количество элементов соответствия |
| ***Надежность*** | | |
| Завершенность | Полнота  тестирования (1) | ***А*** – количество тестовых комбинаций, спроектированных в плане тестирования и подтвержденных при проверке; ***В*** – количество требуемых тестовых комбинаций |
| Устойчивость к ошибке | Предотвращение  некорректных действий (1) | ***А*** – количество функций, реализованных c предотвращением некорректных действий; ***В*** – количество типичных некорректных действий, которое должно быть учтено |
| Восстанавливаемость | Способность к восстановлению  (1) | ***А*** – количество реализованных требований к восстановлению, подтвержденное при проверке; ***В*** – общее количество требований к восстановлению, определенных в спецификации |
| Соответствие  надежности | Соответствие надежности (1) | ***А*** – количество корректно реализованных элементов, связанных с соответствием надежности, подтвержденное при оценке; ***В*** – общее количество элементов соответствия |
| ***Практичность*** | | |
| Понятность | Способность к демонстрации (1) | ***А*** – количество демонстрируемых функций, подтвержденное при проверке; ***В*** – общее количество функций, которые должны обладать способностью к демонстрации |
| Обучаемость | Полнота документации пользователя и/или возможности электронной справки help (1) | ***А*** – количество описанных функций; ***В*** – общее количество предоставляемых функций |
| Простота использования | Отменяемость действий  пользователя (1) | ***А*** – количество реализованных функций, которые могут быть отменены пользователем с восстановлением предыдущих данных; ***В*** – общее количество функций |
| Привлекательность | Настраиваемость вида  интерфейса пользователя (1) | ***А*** – количество типов элементов интерфейса, которые могут быть настроены; ***В*** – общее количество типов элементов интерфейса |
| Соответствие  практичности | Соответствие практичности (1) | ***А*** – количество корректно реализованных элементов, связанных с соответствием практичности, подтвержденное при оценке; ***В*** – общее количество элементов соответствия |
| ***Эффективность*** | | |
| Поведение во времени | Пропускная способность ***Х = А*** | ***А*** – число задач в единицу времени, подтвержденное при проверке |
| Использование ресурсов | Использование памяти  ***Х = А*** | ***А*** – размер памяти в байтах (вычисленный или моделированный) |
| Соответствие эффективности | Соответствие эффективности (1) | ***А*** – количество корректно реализованных элементов, связанных с соответствием эффективности, подтвержденное при проверке; ***В*** – общее количество элементов соответствия |
| ***Сопровождаемость*** | | |
| Анализируемость | Готовность диагностических функций (1) | ***А*** – количество реализованных диагностических функций из заданных в спецификации, подтвержденное при проверке; ***В*** – требуемое количество диагностических функций |
| Изменяемость | Регистрируемость  изменений (1) | ***А*** – количество изменений в функциях/модулях, отраженных в комментариях, подтвержденное при проверке; ***В*** – общее количество изменений в функциях/модулях относительно оригинального кода |
| Стабильность | Влияние изменений  (2) | ***А*** – количество обнаруженных вредных влияний после модификаций; ***В*** – количество сделанных модификаций |
| Тестируемость | Полнота встроенных функций  тестирования (1) | ***А*** – количество реализованных встроенных функций тестирования из заданных в спецификации, подтвержденное при проверке; ***В*** – требуемое количество встроенных функций тестирования |
| Соответствие  сопровождаемости | Соответствие  сопровождаемости  (1) | ***А*** – количество корректно реализованных элементов, связанных с соответствием сопровожаемости, подтвержденное при оценке; ***В*** – общее количество элементов соответствия |
| ***Мобильность*** | | |
| Адаптируемость | Адаптируемость  структур данных  (1) | ***А*** – количество структур данных, работоспособность которых не нарушена после адаптации, подтвержденное при проверке; ***В*** – общее количество структур данных, которые должны обладать способностью к адаптации |
| Настраиваемость | Объем работ по установке  (1) | ***А*** – количество автоматических шагов установки (инсталляции), подтвержденное при проверке; ***В*** – требуемое количество шагов инсталляции |
| Совместимость | Доступная совместимость  (1) | ***А*** – количество объектов, с которыми продукт может сосуществовать, из заданных в спецификации; ***В*** – количество объектов в окружающей среде, с которыми продукт должен сосуществовать |
| Взаимозаменяемость | Преемственность  данных (1) | ***А*** – количество элементов данных ПС, которые продолжают использоваться после замещения (из заданных в спецификации), подтвержденное при проверке; ***В*** – количество элементов старых данных, которые должны использоваться из старого ПС |
| Соответствие  мобильности | Соответствие мобильности (1) | ***А*** – количество корректно реализованных элементов, связанных с соответствием мобильности, подтвержденное при проверке; ***В*** – общее количество элементов соответствия |

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.

### 6.3. 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.

***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.

Табл. 37 содержит примеры внешних метрик качества программных средств из стандарта *ISO/IEC TR 9126–2:2003*. Во втором столбце таблицы по каждой подхарактеристике приведено название одной метрики, уникальная формула или номер формулы (1) или (2) из подразд. 6.1 для оценки данной метрики. Исходные данные в третьем столбце – это данные, используемые в соответствующей формуле для вычисления значения представленной метрики.

Таблица 37

**Внешние метрики качества программных средств**

|  |  |  |
| --- | --- | --- |
| Название  подхарактеристики | Название метрики,  формула для ее оценки | Исходные данные для вычисления метрики  по соответствующей формуле |
| ***Функциональность*** | | |
| Пригодность | Полнота функциональной  реализации (2) | ***А*** – количество отсутствующих функций, обнаруженных при проверке; ***В*** – количество функций, описанных в спецификации требований |
| Правильность | Точность  ***Х=А/Т*** | ***А*** – количество результатов, подсчитанное пользователями, с уровнем точности, отличающимся от требуемого; ***Т*** – продолжительность работы |
| Способность  к взаимодействию | Способность к обмену данными (основанная на успешных попытках пользователя) (2) | ***А*** – количество случаев, в которых пользователю не удалось обменяться данными с другими ПС или системами; ***В*** – количество случаев, в которых пользователь пытался обмениваться данными |
| Защищенность | Предотвращение  разрушения данных  (2) | ***А*** – количество произошедших случаев разрушения важных данных; ***В*** – количество тестовых случаев, направленных на разрушение данных |
| Соответствие функциональности | Соответствие  функциональности  (2) | ***А*** – количество заданных элементов соответствия функциональности, которые не были выполнены при тестировании; ***В*** – общее количество заданных элементов соответствия функциональности |
| ***Надежность*** | | |
| Завершенность | Плотность ошибок  ***Х = А / Z*** | ***А*** – количество ошибок, обнаруженных в течение определенного испытательного срока; ***Z*** – размер продукта |
| Устойчивость к ошибке | Предотвращение  некорректных действий (1) | ***А*** – количество предотвращенных критических и серьезных отказов; ***В*** – количество выполненных при тестировании тестовых случаев, направленных на проверку типовых некорректных действий, которые могут привести к отказу |
| Восстанавливаемость | Способность  к восстановлению (1) | ***А*** – количество случаев успешного восстановления; ***В*** –количество случаев восстановления, протестированных согласно требованиям |
| Соответствие  надежности | Соответствие надежности (2) | ***А*** – количество заданных элементов соответствия надежности, которые не были выполнены при тестировании; ***В*** – общее количество заданных элементов соответствия надежности |
| ***Практичность*** | | |
| Понятность | Полнота описания (1) | ***А*** – количество функций (или классов функций), понятных после прочтения документации на программный продукт; ***В*** – общее количество функций (или классов функций), реализуемых программным продуктом |
| Обучаемость | Эффективность документации пользователя и/или справочной системы (help) (1) | ***А*** – количество задач, успешно выполненных после получения оперативной справки и/или чтения документации; ***В*** – общее количество протестированных задач |
| Простота использования | Исправление ошибок  при использовании (1) | ***А*** – число экранов или форм, где входные данные были успешно модифицированы или изменены (восстановлены) перед очередной обработкой; ***В*** – число экранов или форм, где пользователь пытался модифицировать или изменить (восстановить) входные данные в течение испытательного срока использования |
| Привлекательность | Изменяемость вида  интерфейса (1) | ***А*** – количество элементов интерфейса, измененных внешне для удовлетворения пользователя; ***В*** –количество элементов интерфейса, которые пользователь хочет изменить |
| Соответствие  практичности | Соответствие практичности (2) | ***А*** – количество заданных элементов соответствия практичности, которые не были выполнены при тестировании; ***В*** – общее количество заданных элементов соответствия практичности |
| ***Эффективность*** | | |
| Поведение во времени | Время отклика ***Х = А – В*** | ***А*** – момент времени получения результата; ***В*** – момент времени завершения ввода команды |
| Использование ресурсов | Использование устройств  ввода/вывода (1) | ***А*** – время занятости устройств ввода/вывода;  ***В*** – заданное время, предназначенное для использования устройств ввода/вывода |
| Соответствие эффективности | Соответствие эффективности (2) | ***А*** – количество заданных элементов соответствия эффективности, которые не были выполнены при тестировании; ***В*** – общее количество заданных элементов соответствия эффективности |
| ***Сопровождаемость*** | | |
| Анализируемость | Поддержка  диагностическими функциями (1) | ***А*** – количество отказов, при которых персонал сопровождения с помощью диагностических функций может диагностировать причину; ***В*** – общее число зарегистрированных отказов |
| Изменяемость | Возможность управления  изменением ПС (1) | ***А*** – количество фактически записанных данных регистрации изменений; ***В*** – запланированное количество данных регистрации изменений, достаточное для отслеживания изменений ПС |
| Стабильность | Возникновение отказа после изменения  ***X = A / N*** | ***А*** – количество отказов, возникших в течение заданного периода после устранения отказа; ***N*** – количество отказов, устраненных путем изменения ПС |
| Тестируемость | Доступность  встроенных функций  тестирования (1) | ***А*** – количество случаев, в которых персонал сопровождения может использовать встроенные функции тестирования; ***В*** – количество подходящих случаев, в которых можно было бы использовать встроенные тесты |
| Соответствие  сопровождаемости | Соответствие  сопровождаемости  (2) | ***А*** – количество заданных элементов соответствия сопровождаемости, которые не были выполнены при тестировании; ***В*** – общее количество заданных элементов соответствия сопровождаемости |
| ***Мобильность*** | | |
| Адаптируемость | Адаптируемость  структур данных  (1) | ***А*** – количество работоспособных данных, которые не требуют сопровождения при адаптации; ***В*** – ожидаемое число работоспособных данных в окружающей среде, к которой ПС адаптировано |
| Настраиваемость | Простота установки  (1) | ***А*** – число успешных случаев приспосабливания пользователем операции инсталляции к среде эксплуатации; ***В*** – общее число попыток пользователя приспособить операцию инсталляции к среде окружения |
| Совместимость | Доступная совместимость  ***Х = А / Т*** | ***А*** – число любых ограничений или непредусмотренных отказов, с которыми пользователь сталкивается во время одновременной работы с другими ПС; ***Т*** – продолжительность одновременной работы с другими ПС |
| Взаимозаменяемость | Преемственность данных (1) | ***А*** – число данных замещаемого ПС, которые могут продолжать использоваться после его замещения; ***В*** – число данных замещаемого ПС, которые по плану должны продолжать использоваться после его замещения |
| Соответствие  мобильности | Соответствие мобильности (2) | ***А*** – количество заданных элементов соответствия мобильности, которые не были выполнены при тестировании; ***В*** – общее количество заданных элементов соответствия мобильности |

Не все метрики, приведенные в данной таблице, 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 (3) (см. подразд. 6.1). Это затрудняет их использование при интегральной оценке качества программных продуктов.

### 6.4. 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.

Табл. 38 содержит примеры метрик качества в использовании из стандарта *ISO/IEC TR 9126–4:2004*. Во втором столбце таблицы по каждой подхарактеристике приведено название одной метрики, уникальная формула или номер формулы (1) или (2) из подразд. 6.1 для оценки данной метрики. Исходные данные в третьем столбце – это данные, используемые в соответствующей формуле для вычисления значения представленной метрики.

Таблица 38

Метрики качества ПС в использовании

|  |  |  |
| --- | --- | --- |
| Название  характеристики | Название метрики,  формула для ее оценки | Исходные данные для вычисления метрики  по соответствующей формуле |
| Результативность | Завершение  задачи (1) | ***А*** – количество завершенных задач; ***В*** – общее количество попыток выполнения задач |
| Продуктивность | Коэффициент  продуктивности (1) | ***А*** – продуктивное время; ***А*** **=** ***В*** ***– (В1 + В2 + В3)***, где ***В*** – продолжительность выполнения задачи; ***В1*** – продолжительность помощи; ***В2*** – продолжительность обработки ошибок; ***В3*** – продолжительность поиска |
| Безопасность | Экономический  ущерб (2) | ***А*** – число случаев экономического ущерба; ***В*** – общее число случаев использования |
| Удовлетворенность | Использование  по собственному  усмотрению (1) | ***А*** – количество случаев использования заданных в спецификации функций программного средства/приложений/систем; ***В*** – количество случаев их запланированного использования |

## 7. Сертификация программных средств

### 7.1. Основные понятия и определения в области технического нормирования, стандартизации и оценки соответствия

5 января 2004 г. в Республике Беларусь приняты и введены в действие *Законы № 262-З «О техническом нормировании и стандартизации» и № 269-З «Об оценке соответствия требованиям технических нормативных правовых актов в области технического нормирования и стандартизации»* [, ].

Данные законы разработаны с учетом соглашений Всемирной торговой организации (ВТО). Они базируются на практике Европейского Союза (ЕС), определенной в Директивах ЕС Нового и Глобального подхода [, ], и направлены на обеспечение единой государственной политики в области технического нормирования, стандартизации и оценки соответствия.

Первый из законов 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.

Второй закон 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.

В Законе «О техническом нормировании и стандартизации» определены следующие ***технические нормативные правовые акты*** в области технического нормирования и стандартизации:

* технические регламенты;
* технические кодексы установившейся практики;
* государственные стандарты Республики Беларусь;
* технические условия.

***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.

***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.

***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.

***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.

В Законе [] 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*** [].

*Аккредитация:* вид оценки соответствия, результатом осуществления которого является официальное признание компетентности юридического лица в выполнении работ по подтверждению соответствия и (или) проведении испытаний продукции.

*Аккредитованная испытательная лаборатория (центр):* юридическое лицо, аккредитованное для проведения испытаний продукции в определенной области аккредитации.

*Аккредитованный орган по сертификации:* юридическое лицо, аккредитованное для выполнения работ по подтверждению соответствия в определенной области аккредитации.

*Аттестат аккредитации:* документ, удостоверяющий компетентность юридического лица в выполнении работ по подтверждению соответствия или в проведении испытаний продукции в определенной области аккредитации.

*Декларация о соответствии:* документ, в котором изготовитель (продавец) удостоверяет соответствие производимой и (или) реализуемой им продукции требованиям технических нормативных правовых актов в области технического нормирования и стандартизации.

*Декларирование соответствия:* форма подтверждения соответствия, осуществляемого изготовителем (продавцом).

*Национальная система подтверждения соответствия Республики Беларусь:* установленная совокупность субъектов оценки соответствия, нормативных правовых актов и технических нормативных правовых актов в области технического нормирования и стандартизации, определяющих правила и процедуры подтверждения соответствия и функционирования 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 продукции.

*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 требованиям технических нормативных правовых актов в области технического нормирования и стандартизации.

*Сертификат компетентности:* документ, удостоверяющий профессиональную компетентность физического лица в выполнении определенных работ, услуг.

*Сертификат соответствия:* документ, удостоверяющий соответствие объекта оценки требованиям технических нормативных правовых актов в области технического нормирования и стандартизации.

*Сертификация:* форма подтверждения соответствия, осуществляемого аккредитованным органом по сертификации.

*Система управления качеством (система менеджмента качества):* часть общей системы управления, включающая организационную структуру, планирование, ответственность, методы, процедуры, процессы, ресурсы, необходимые для обеспечения качества продукции и (или) услуг; как правило, система управления качеством является частью системы управления предприятием или организацией.

*Схема подтверждения соответствия:* установленная последовательность действий, результаты которых рассматриваются в качестве доказательств соответствия объекта оценки требованиям технических нормативных правовых актов в области технического нормирования и стандартизации.

*Форма подтверждения соответствия:* установленный порядок документального удостоверения соответствия объекта оценки требованиям технических нормативных правовых актов в области технического нормирования и стандартизации.

### 7.2. Общие сведения об оценке соответствия в Республике Беларусь

Оценка соответствия в Беларуси выполняется на основе *Закона Республики Беларусь № 269-З «Об оценке соответствия требованиям технических нормативных правовых актов в области технического нормирования и стандартизации»* []. Оценка соответствия осуществляется *в целях*:

* обеспечения защиты жизни, здоровья и наследственности человека, имущества и охраны окружающей среды;
* повышения конкурентоспособности продукции (услуг);
* создания благоприятных условий для обеспечения свободного перемещения продукции на внутреннем и внешнем рынках.

*Принципами* оценки соответствия являются:

* гармонизация с международными и межгосударственными (региональными) подходами в области оценки соответствия;
* обеспечение идентичности правил и процедур подтверждения соответствия продукции отечественного и иностранного производства;
* соблюдение требований конфиденциальности сведений, полученных при выполнении работ по оценке соответствия.
* Основными объектами оценки соответствия являются:
* продукция;
* процессы жизненного цикла продукции (разработка, производство, эксплуатация, хранение, перевозка, реализация и утилизация);
* оказание услуг;
* система управления качеством;
* система управления окружающей средой;
* юридические лица;
* персонал.
* Субъектами оценки соответствия являются:
* государство в лице уполномоченных государственных органов;
* аккредитованные органы по сертификации;
* аккредитованные испытательные лаборатории (центры);
* заявители на аккредитацию;
* заявители на подтверждение соответствия.

Законом [] предусмотрено *два вида оценки соответствия*: аккредитация и подтверждение соответствия (рис. 41).

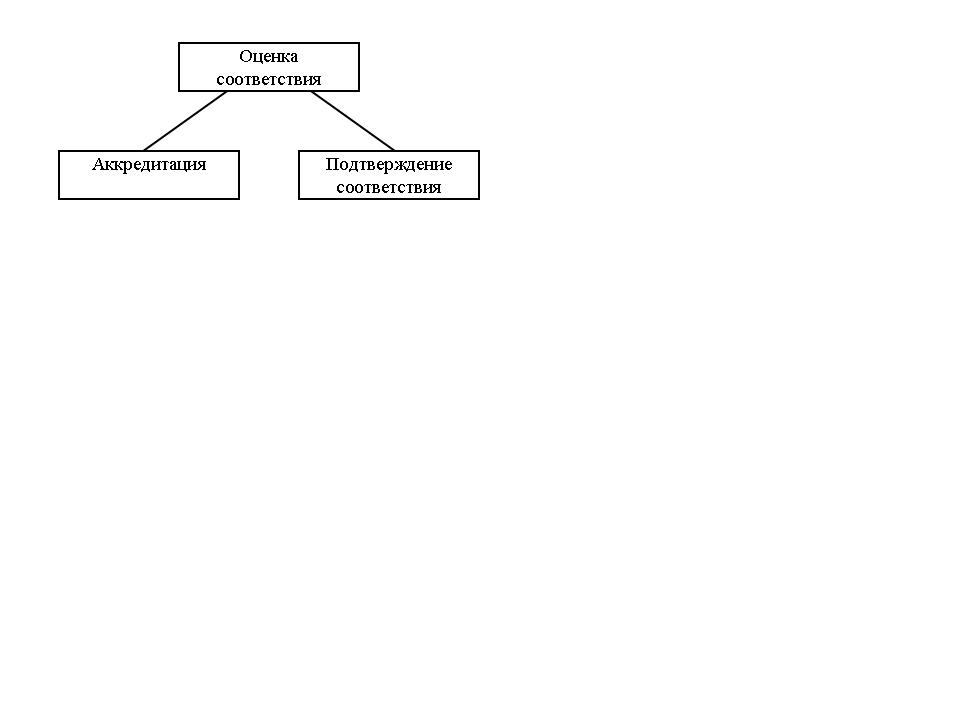


Рис. . Виды оценки соответствия

К *документам* об оценке соответствия относятся:

* аттестат аккредитации;
* сертификат соответствия;
* декларация о соответствии;
* сертификат компетентности.

Систему подтверждения соответствия в Республике Беларусь иллюстрирует рис. 42 [].

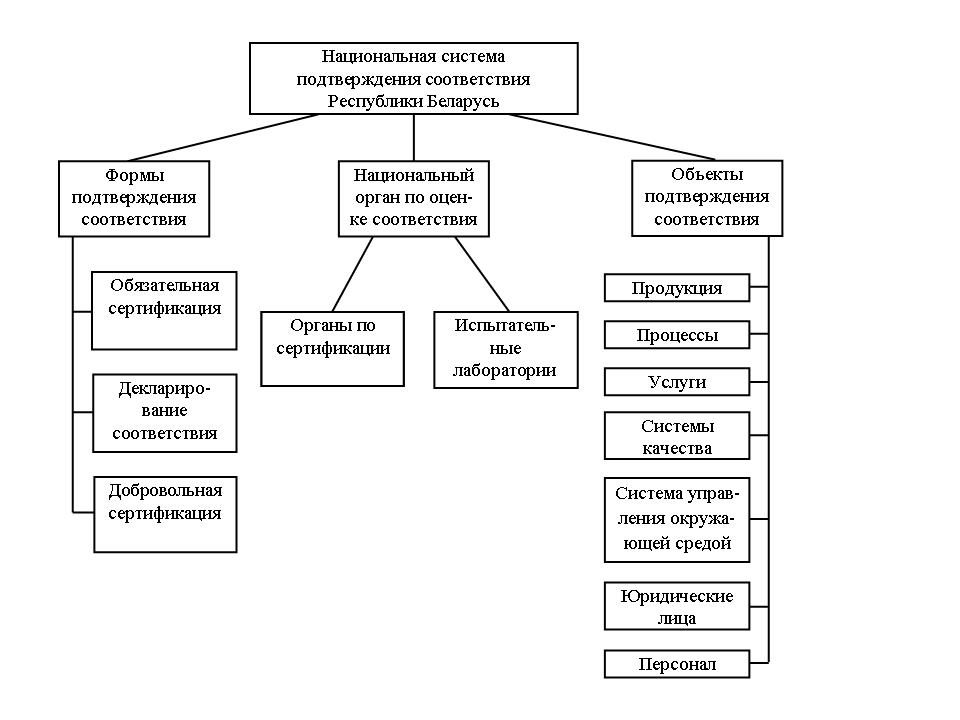


Рис. . Система подтверждения соответствия в Республике Беларусь

Национальным органом по оценке соответствия Республики Беларусь является Государственный комитет по стандартизации, метрологии и сертификации при Совете министров Республики Беларусь (Госстандарт). К органам по сертификации относится, например, Белорусский государственный институт стандартизации и сертификации (БелГИСС).

Подтверждение соответствия осуществляется в *целях*:

* удостоверения соответствия объектов оценки требованиям технических нормативных правовых актов в области технического нормирования и стандартизации;
* содействия потребителям в компетентном выборе продукции (услуг).

*Принципами подтверждения соответствия* являются:

* открытость и доступность правил и процедур подтверждения соответствия;
* независимость аккредитованных органов по сертификации и аккредитованных испытательных лабораторий (центров) от заявителей на подтверждение соответствия;
* минимизация сроков исполнения и затрат на прохождение процедур подтверждения соответствия;
* защита имущественных интересов заявителей на подтверждение соответствия, в том числе путем соблюдения конфиденциальности в отношении сведений, полученных в процессе прохождения процедур подтверждения соответствия;
* недопустимость ограничения конкуренции при выполнении работ по подтверждению соответствия и проведении испытаний продукции.

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 других объектов оценки соответствия, подлежащих обязательному подтверждению соответствия в Республике Беларусь. В данном Перечне устанавливаются:

* виды продукции, услуг, персонал 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, услуг, деятельности персонала и функционирования других объектов оценки соответствия для жизни, здоровья и наследственности человека, имущества и окружающей среды;
* несовместимость технических средств отечественного и иностранного производства.

***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 *только в отношении продукции* 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.

*Форма, правила и процедуры* обязательного подтверждения соответствия, а также правила и процедуры добровольной сертификации устанавливаются в документах Национальной системы подтверждения соответствия Республики Беларусь (НСПС РБ).

К *основополагающим документам* НСПС РБ относятся:

* *ТКП 5.1.01–2004 НСПСРБ. Основные положения.* Данный ТКП является основным в комплексе ТКП и стандартов, обеспечивающих функционирование НСПС РБ. ТКП определяет общие правила организации работ по подтверждению соответствия, структуру НСПС РБ и функции ее органов.
* *ТКП 5.1.02–2004 НСПС РБ. Порядок сертификации продукции. Основные положения.* ТКП устанавливает общие требования к порядку проведения сертификации отечественной и импортируемой продукции и применяется при обязательной и добровольной сертификации продукции. На его основе разрабатываются и применяются совместно с ним порядки сертификации групп однородной продукции, учитывающие особенности ее ЖЦ.
* *ТКП 5.1.03–2004 НСПС РБ. Порядок декларирования соответствия продукции. Основные положения.* ТКП устанавливает порядок проведения декларирования соответствия отечественной и импортируемой продукции требованиям технических нормативных правовых актов в области технического нормирования и стандартизации.
* *ТКП 5.1.04–2004 НСПС РБ. Порядок сертификации услуг. Основные положения.* ТКП устанавливает общие требования к порядку сертификации услуг в рамках НСПС РБ. Данный ТКП применяется при проведении сертификации услуг и разработке порядка сертификации конкретного вида услуг.
* *ТКП 5.1.05–2004 НСПС РБ. Порядок сертификации систем менеджмента качества. Основные положения.* ТКП устанавливает общие требования к порядку проведения работ по сертификации систем менеджмента качества на соответствие требованиям технических нормативных правовых актов (СТБ ИСО 9001 и др.). Требования ТКП могут использоваться органами по сертификации для разработки порядка проведения сертификации систем менеджмента качества организаций.
* *ТКП 5.1.06–2004 НСПС РБ. Порядок сертификации компетентности персонала. Основные положения.* ТКП устанавливает общие требования к порядку проведения сертификации профессиональной компетентности персонала, осуществляющего конкретные виды работ (услуг) в определенной области деятельности. Требования ТКП применяются органами по сертификации при разработке порядка проведения сертификации персонала.
* *ТКП 5.1.09–2004 НСПС РБ. Порядок сертификации экспертов–аудиторов по качеству.* ТКП устанавливает требования, предъявляемые к экспертам–аудиторам по качеству в НСПС РБ, права, обязанности, ответственность и порядок их сертификации.
* *ТКП 5.1.11–2004 НСПС РБ. Порядок применения форм и схем подтверждения соответствия требованиям технических нормативных правовых актов в области технического нормирования и стандартизации.* ТКП устанавливает порядок выбора и применения форм и схем обязательного подтверждения соответствия продукции требованиям технических нормативных правовых актов при разработке технических регламентов.

### 7.3. 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 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*. Общий процесс оценки, регламентированный в данных стандартах, подробно рассмотрен в подразд. 5.5.

## Литература

**Основная и дополнительная литература**

1. Бахтизин, В. В. Стандартизация и сертификация программного обеспечения : учеб. пособие / В. В. Бахтизин, Л. А. Глухова. – Минск : БГУИР, 2006.
2. ГОСТ 28195–89. Оценка качества программных средств. Общие положения.
3. ГОСТ 28195–99. Оценка качества программных средств. Общие положения.
4. ГОСТ 28806–90. Качество программных средств. Термины и определения.
5. ГОСТ Р ИСО/МЭК 12207–99. Информационная технология. Процессы жизненного цикла программных средств.
6. ГОСТ Р ИСО/МЭК ТО 15271–2002. Информационная технология. Руководство по применению ГОСТ Р ИСО/МЭК 12207 (Процессы жизненного цикла программных средств).
7. ГОСТ Р ИСО/МЭК ТО 16326–2002. Программная инженерия. Руководство по применению ГОСТ Р ИСО/МЭК 12207 при управлении проектом.
8. СТБ ИСО/МЭК 12207–2003. Информационная технология. Процессы жизненного цикла программных средств.
9. СТБ ИСО/МЭК 9126–2003. Информационные технологии. Оценка программной продукции. Характеристики качества и руководства по их применению.
10. ISO 10006–1997. Управление качеством – Руководства по качеству при управлении проектом.
11. ISO/IEC 12207:1995. Информационная технология – Процессы жизненного цикла программных средств.
12. ISO/IEC 12207:2008. Системная и программная инженерия– Процессы жизненного цикла программных средств.
13. ISO/IEC 14598***–***1:1999. Информационная технология – Оценка программного продукта – Часть 1: Общий обзор.
14. ISO/IEC 14598***–***2:2000. Программная инженерия – Оценка продукта – Часть 2: Планирование и управление.
15. ISO/IEC 14598***–***3:2000. Программная инженерия – Оценка продукта – Часть 3: Процесс для разработчиков.
16. ISO/IEC 14598–4:1999. Программная инженерия – Оценка продукта – Часть 4: Процесс для заказчиков.
17. ISO/IEC 14598–5:1998. Информационная технология – Оценка программного продукта – Часть 5: Процесс для оценщиков.
18. ISO/IEC 14598–6:2001. Программная инженерия – Оценка продукта – Часть 6: Документация модулей оценки.
19. ISO/IEC 15939:2002. Программная инженерия – Процесс измерения программных средств.
20. ISO/IEC 2382–1:1993. Информационные технологии – Словарь – Часть 1: Основополагающие термины.
21. ISO/IEC 2382–20:1990. Информационные технологии – Словарь – Часть 20: Разработка систем.
22. ISO 8402:1994. Управление качеством и обеспечение качества. Словарь.
23. ISO/IEC 9126–1:2001. Программная инженерия – Качество продукта – Часть 1: Модель качества.
24. ISO/IEC TR 9126–2:2003. Программная инженерия – Качество продукта – Часть 2: Внешние метрики.
25. ISO/IEC TR 9126–3:2003. Программная инженерия – Качество продукта – Часть 3: Внутренние метрики.
26. ISO/IEC TR 9126–4:2004. Программная инженерия – Качество продукта – Часть 4: Метрики качества в использовании.
27. Руководство РМВОК. Руководство по сведениям для органа управления проектом. – Комитет по стандартам института управления проектом PMI (США). – 1996.
28. Закон Республики Беларусь от 5 января 2004 г. № 262-З «О техническом нормировании и стандартизации».
29. Закон Республики Беларусь от 5 января 2004 г. № 269-З «Об оценке соответствия требованиям технических нормативных правовых актов в области технического нормирования и стандартизации».
30. Council Resolution of 7 May 1985 on a new approach to technical harmonization and standards.
31. Council Resolution of 21 December 1989 on a global approach to conformity assessment.
32. Липаев В.В. Обеспечение качества программных средств. Методы и стандарты. – М.: СИНТЕГ, 2001.
33. [www.gosstandart.gow.by](http://www.gosstandart.gow.by/)

## Вопросы для самопроверки по теоретическому разделу дисциплины «Метрология, стандартизация и сертификация в информатике и радиоэлектронике»

**Раздел 1**

1. Дайте определение жизненного цикла (ЖЦ) программного средства (ПС) или системы.
2. Дайте определение модели ЖЦ ПС или системы.
3. Что такое аттестация ПС?
4. Что такое верификация ПС?
5. Опишите понятие базовой линии.
6. Дайте определение элемента конфигурации.

**Раздел 2**

1. Назовите базовый стандарт в области ЖЦ ПС и систем.
2. Что подразумевается под системой при стандартизации ЖЦ?
3. Определите иерархическую структуру ЖЦ ПС.
4. На какие группы делятся процессы ЖЦ и в соответствии с чем?
5. Назовите основные стороны, участвующие в ЖЦ ПС и систем.
6. Перечислите процессы ЖЦ в каждой группе.
7. Опишите сущность каждого из основных процессов ЖЦ.
8. Назовите процессы ЖЦ, влияющие на управление качеством ПС.
9. Перечислите работы процесса разработки и опишите их содержание.
10. В каком процессе ЖЦ разрабатывается план управления проектом?
11. Перечислите содержание плана управления проектом.
12. В каком процессе ЖЦ разрабатываются планы выполнения процессов?
13. Опишите содержание планов выполнения процессов.
14. Перечислите состав требований к системе и назовите критерии их оценки.
15. Перечислите состав требований к ПС и назовите критерии их оценки.
16. Определите структуру работ по проектированию ПС и их результаты.
17. Назовите критерии оценки результатов каждой работы процесса разработки.
18. Назовите основные документы процесса разработки.
19. Опишите сущность каждого из вспомогательных процессов ЖЦ.
20. Опишите сущность каждого из организационных процессов ЖЦ.
21. Что такое инфраструктура процесса?
22. Опишите сущность процесса адаптации положений стандарта *СТБ* *ИСО/МЭК 12207-2003* к условиям конкретного проекта.
23. Какие характеристики проекта влияют на адаптацию?
24. Охарактеризуйте назначение основных инструментальных средств линейки интегрированных продуктов *Telelogic*.

**Раздел 3**

1. Назовите обобщенные показатели программных средств, существенные при их разработке и оценке качества.
2. В чем различия внутреннего, внешнего качества и качества в использовании?
3. Что такое контекст использования?
4. Дайте понятие метрики, назовите и опишите виды метрик.
5. Дайте определение оценки и модели качества.
6. Определите понятие ранжирования и уровня ранжирования.
7. Дайте определение уровня качества функционирования.
8. Дайте определение шкалы, перечислите и опишите типы шкал.

**Раздел 4**

1. Назовите стандарты в области оценки качества, действующие в Беларуси.
2. Классифицируйте и опишите методы определения показателей качества ПС.
3. Опишите различия иерархических моделей качества программных средств, определенных в ГОСТ 28195–99 и СТБ ИСО/МЭК 9126–2003.
4. Перечислите факторы и критерии качества ПС по ГОСТ 28195–99.
5. Опишите модель Надежности, регламентированную в ГОСТ 28195–99.
6. Опишите метод оценки качества ПС, рекомендованный в ГОСТ 28195–99.
7. Опишите метод оценки качества программных средств, регламентированный в СТБ ИСО/МЭК 9126–2003.
8. Приведите примеры диапазонов ранжирования измеренных значений.

**Раздел 5**

1. Опишите содержание каждой из частей серии стандартов ISO/IEC 9126–1–4.
2. Опишите содержание каждой части серии стандартов ISO/IEC 14598–1–6.
3. Опишите и представьте графически связь качества ПС с его 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 каждую из характеристик модели внутреннего и внешнего качества программных средств, определенной в ISO/IEC 9126–1:2001.
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 6**

1. Какие международные стандарты определяют метрики качества программных средств?
2. Перечислите и опишите желательные свойства метрик качества ПС.
3. Перечислите и опишите критерии обоснованности метрик качества ПС.
4. Какие типы шкал обычно используются при измерении значений метрик качества ПС?
5. Какие типы измеренного значения обычно используются при измерении значений метрик качества ПС?
6. По каким формулам рекомендуется вычислять значения метрик качества ПС?
7. Поясните, для чего рекомендуется представлять значения метрик качества ПС в относительных единицах.
8. В каком диапазоне значений рекомендуется представлять значения метрик качества ПС?
9. Определите назначение внутренних метрик качества ПС.
10. Определите назначение внешних метрик качества ПС.
11. Определите назначение метрик качества ПС в использовании.
12. По каждой характеристике качества приведите примеры внутренних метрик качества ПС.
13. По каждой характеристике качества приведите примеры внешних метрик качества ПС.
14. По каждой характеристике качества приведите примеры метрик качества в использовании.

**Раздел 7**

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 их 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.

## Вопросы к зачету по дисциплине «Метрология, стандартизация и сертификация в информатике и радиоэлектронике»

1. Основные понятия в области жизненного цикла программных средств. Компоненты жизненного цикла. Иерархическая структура жизненного цикла.
2. СТБ ИСО/МЭК 12207-2003. Процессы жизненного цикла программных средств и их классификация.
3. СТБ ИСО/МЭК 12207-2003. Основные процессы жизненного цикла.
4. СТБ ИСО/МЭК 12207-2003.Процесс заказа и его структура.
5. СТБ ИСО/МЭК 12207-2003. Процесс поставки и его структура.
6. СТБ ИСО/МЭК 12207-2003. Процесс разработки и его структура.
7. СТБ ИСО/МЭК 12207-2003. Процесс эксплуатации и его структура.
8. СТБ ИСО/МЭК 12207-2003. Процесс сопровождения и его структура.
9. СТБ ИСО/МЭК 12207-2003. Вспомогательные процессы жизненного цикла.
10. СТБ ИСО/МЭК 12207-2003. Процесс документирования и его структура.
11. СТБ ИСО/МЭК 12207-2003. Процесс управления конфигурацией и его структура.
12. СТБ ИСО/МЭК 12207-2003. Процесс обеспечения качества и его структура.
13. СТБ ИСО/МЭК 12207-2003. Процесс верификации и его структура.
14. СТБ ИСО/МЭК 12207-2003. Процесс аттестации и его структура.
15. СТБ ИСО/МЭК 12207-2003. Процесс совместного анализа и его структура.
16. СТБ ИСО/МЭК 12207-2003. Процесс аудита и его структура.
17. СТБ ИСО/МЭК 12207-2003. Процесс решения проблем и его структура.
18. СТБ ИСО/МЭК 12207-2003. Организационные процессы жизненного цикла.
19. СТБ ИСО/МЭК 12207-2003. Процесс управления и его структура.
20. СТБ ИСО/МЭК 12207-2003. Процесс создания инфраструктуры и его структура.
21. СТБ ИСО/МЭК 12207-2003. Процесс усовершенствования и его структура.
22. СТБ ИСО/МЭК 12207-2003. Процесс обучения и его структура.
23. СТБ ИСО/МЭК 12207-2003. Процессы, связанные с обеспечением качества программных средств.
24. Адаптация требований стандарта СТБ ИСО/МЭК 12207-2003к условиям конкретного проекта.
25. Инструментальные средства автоматизации жизненного цикла программных средств.
26. Качество программных средств. Основные понятия и определения. Показатели качества программных средств. Метрики и их типы. Шкала. Типы шкал. Атрибуты.
27. Общие сведения о стандартах в области оценки качества, действующих на территории Республики Беларусь.
28. ГОСТ 28195-99. Методы определения показателей качества программных средств и их классификация.
29. ГОСТ 28195-99. Иерархическая модель оценки качества программного средства.
30. ГОСТ 28195-99. Иерархическая модель сопровождаемости.
31. ГОСТ 28195-99. Метод количественной оценки качества программных средств.
32. СТБ ИСО/МЭК 9126-2003. Метод оценки качества программных средств.
33. Стандарты серии ISO/IEC в области оценки качества программных средств.
34. Структура серии стандартов ISO/IEC 9126-1-4.
35. Структура серии стандартов ISO/IEC 14598-1-6.
36. ISO/IEC 9126-1:2001. Связь качества программных средств с их жизненным циклом.
37. ISO/IEC 9126-1. Модель внешнего и внутреннего качества программных средств. Характеристики качества.
38. ISO/IEC 9126-1. Функциональность. Подхарактеристики Функциональности.
39. ISO/IEC 9126-1. Надежность. Подхарактеристики Надежности.
40. ISO/IEC 9126-1. Эффективность. Подхарактеристики Эффективности**.**
41. ISO/IEC 9126-1. Практичность. Подхарактеристики Практичности.
42. ISO/IEC 9126-1. Сопровождаемость. Подхарактеристики Сопровождаемости.
43. ISO/IEC 9126-1. Мобильность. Подхарактеристики Мобильности.
44. ISO/IEC 9126-1. Модель качества в использовании. Характеристики качества в использовании.
45. ISO/IEC 14598-1. Метод оценки качества программных средств.
46. ISO/IEC 9126. Свойства и критерии обоснованности метрик качества программных средств.
47. ISO/IEC 9126-3. Внутренние метрики качества программных средств.
48. ISO/IEC 9126-2. Внешние метрики качества программных средств.
49. ISO/IEC 9126-4. Метрики качества программных средств в использовании.
50. Основные понятия сертификации программных средств и систем качества. Цели сертификации.
51. Основные понятия в области технического нормирования, стандартизации и оценки соответствия.
52. Оценка соответствия в Республике Беларусь.
53. Организация сертификации программных средств.

# Практический раздел

## Контрольные работы

### Контрольная работа № 1

Тема задания

Жизненный цикл программных средств и оценка их качества. Для заданного преподавателем варианта выполнить задания, связанные с жизненным циклом программных средств и систем и оценкой их качества.

Методические указания к выполнению контрольной работы

К выполнению контрольной работы № 1 нужно приступать после изучения следующих тем теоретического раздела: тема 1 (Жизненный цикл программных средств и систем. Основные понятия и определения), 2 (СТБ ИСО/МЭК 12207-2003 – базовый стандарт Беларуси в области жизненного цикла программных средств и систем), 3 (Качество программных средств. Основные понятия и определения), 4 (Стандартизация качества программного обеспечения в Республике Беларусь), 5 (Стандартизация качества программных средств за рубежом), 7 (Сертификация программных средств).

**Контрольную работу следует выполнить и выслать не позднее, чем за две недели до очередной экзаменационной сессии.**

Содержание отчета по контрольной работе № 1

Контрольная работа №1 должна содержать:

* титульный лист (образец титульного листа приведен ниже);
* перечень заданий заданного варианта;
* подробные ответы по каждому из заданий.

Указания к выбору варианта контрольной работы № 1

Номер варианта индивидуального задания определяется следующим образом: 1 плюс остаток от целочисленного деления номера вашей зачетной книжки (его двух последних цифр, стоящих после дефиса) на тридцать пять.

**Например**, номер варианта студента, имеющего номер зачетки 801021-25, равен 1 + 25 mod 35 = 26; номер варианта студента, имеющего номер зачетки 801021-48, равен: 1 + 48 mod 35 = 14; номер варианта студента, имеющего номер зачетки 801021-35, равен: 1 + 35 mod 35 = 1.

Варианты индивидуальных заданий по контрольной работе № 1

1. а) На основании положений СТБ ИСО/МЭК 12207-2003 выполните сравнительный анализ понятий системы и программного средства.

б) Дайте сравнительную характеристику методов оценки качества программных средств, регламентированных в СТБ ИСО/МЭК 9126–2003 и в ГОСТ 28195–99.

1. а) На основании положений СТБ ИСО/МЭК 12207-2003 опишите общие черты и различия процесса разработки программного средства и процесса разработки системы. Какие работы процесса разработки не используются, если разрабатывается не система, а только программное средство?

б) Перечислите методы определения показателей качества программных средств и дайте их сравнительную характеристику.

1. а) На основании положений СТБ ИСО/МЭК 12207-2003 выполните сравнительный анализ работ процесса разработки, связанных с различными этапами проектирования программных средств, и результатов данных работ.

б) Исходя из положений ISO/IEC 9126-1:2001 дайте сравнительную характеристику моделей внутреннего и внешнего качества программных средств.

1. а) На основании положений СТБ ИСО/МЭК 12207-2003 выполните сравнительный анализ критериев оценки результатов тринадцати работ процесса разработки. Определите связи между критериями различных работ.

б) Перечислите типы шкал, используемых при оценке качества программных средств, и дайте их сравнительную характеристику.

1. а) На основании положений СТБ ИСО/МЭК 12207-2003 выполните сравнительный анализ понятий и процессов аттестации и верификации программных средств и систем.

б) Базируясь на положениях ISO/IEC 9126-1:2001 дайте сравнительную характеристику моделей внутреннего качества и качества в использовании программных средств.

1. а) На основании положений СТБ ИСО/МЭК 12207-2003 выполните сравнительный анализ понятий и процессов аттестации и аудита жизненного цикла программных средств и систем.

б) Исходя из положений ISO/IEC 9126-1:2001 дайте сравнительную характеристику моделей внешнего качества и качества в использовании программных средств.

1. а) На основании положений СТБ ИСО/МЭК 12207-2003 выполните сравнительный анализ процессов совместного анализа и решения проблем жизненного цикла программных средств и систем.

б) Опишите общие черты и различия иерархических моделей качества программных средств, определенных в ГОСТ 28195–99 и СТБ ИСО/МЭК 9126–2003.

1. а) На основании положений СТБ ИСО/МЭК 12207-2003 выполните сравнительный анализ процессов поставки и управления жизненного цикла программных средств и систем.

б) Опишите общие черты и различия иерархических моделей сопровождаемости программных средств, определенных в ГОСТ 28195–99 и ISO/IEC 9126-1:2001.

1. а) На основании положений СТБ ИСО/МЭК 12207-2003 выполните сравнительный анализ процессов создания инфраструктуры и усовершенствования жизненного цикла программных средств и систем.

б) Опишите общие черты и различия иерархических моделей функциональности программных средств, определенных в ГОСТ 28195–99 и ISO/IEC 9126-1:2001.

1. а) На основании положений СТБ ИСО/МЭК 12207-2003 выполните сравнительный анализ понятий и процессов совместного анализа и аудита жизненного цикла программных средств и систем.

б) Опишите общие черты и различия иерархических моделей эффективности программных средств, определенных в ГОСТ 28195–99 и ISO/IEC 9126-1:2001.

1. а) Дайте сравнительную характеристику понятий программного продукта и программного средства.

б) Выполните сравнительный анализ методов оценки качества программных средств, регламентированных в ГОСТ 28195–99 и в ISO/IEC 14598-1:1999.

1. а) Определите связь между понятиями базовой линии и элемента конфигурации. На основании положений СТБ ИСО/МЭК 12207-2003 охарактеризуйте процесс жизненного цикла, управляющий данными понятиями.

б) Выполните сравнительный анализ методов оценки качества программных средств, регламентированных в СТБ ИСО/МЭК 9126–2003 и в ISO/IEC 14598-1:1999.

1. а) На основании положений СТБ ИСО/МЭК 12207-2003 определите связи между процессами аттестации, верификации, обеспечения качества жизненного цикла программных средств и систем.

б) Опишите общие черты и различия иерархической модели универсальности программных средств, определенной в ГОСТ 28195–99, и иерархической модели мобильности программных средств, определенной в ISO/IEC 9126-1:2001.

1. а) На основании положений СТБ ИСО/МЭК 12207-2003 определите связи между процессами разработки и сопровождения программных средств. Опишите различия данных процессов.

б) Перечислите виды технических нормативных правовых актов, определенных в законе РБ № 262-З «О техническом нормировании и стандартизации», и дайте их сравнительную характеристику.

1. а) На основании положений СТБ ИСО/МЭК 12207-2003 определите связи между процессами заказа и поставки программных средств и систем. Опишите различия данных процессов.

б) Перечислите виды качества программных средств, дайте их сравнительную характеристику и опишите их взаимосвязь в жизненном цикле.

1. а) На основании положений СТБ ИСО/МЭК 12207-2003 определите связи между процессами эксплуатации и сопровождения программных средств. Опишите различия данных процессов.

б) Перечислите виды документов об оценке соответствия, дайте сравнительную характеристику данных документов.

1. а) На основании положений СТБ ИСО/МЭК 12207-2003 охарактеризуйте процессы жизненного цикла, влияющие на управление качеством программных средств.

б) Назовите виды оценки соответствия, опишите их общие черты и различия.

1. а) На основании положений СТБ ИСО/МЭК 12207-2003 выполните сравнительный анализ состава требований к системе и состава требований к программным средствам.

б) Опишите общие черты и различия иерархических моделей надежности программных средств, определенных в ГОСТ 28195–99 и ISO/IEC 9126-1:2001.

1. а) На основании положений СТБ ИСО/МЭК 12207-2003 дайте сравнительную характеристику критериев оценки требований к системе и критериев оценки требований к программным средствам.

б) Опишите общие черты и различия иерархической модели удобства использования программных средств, определенной в ГОСТ 28195–99, и иерархической модели практичности программных средств, определенной в ISO/IEC 9126-1:2001.

1. а) На основании положений СТБ ИСО/МЭК 12207-2003 дайте сравнительную характеристику понятий основных, вспомогательных и организационных процессов жизненного цикла программных средств и систем.

б) Опишите общие черты и различия иерархических моделей качества программных средств, определенных в ГОСТ 28195–99 и СТБ ИСО/МЭК 9126–2003.

1. а) Дайте сравнительную характеристику структур жизненного цикла программных средств, регламентированных стандартами СТБ ИСО/МЭК 12207-2003 и ГОСТ 28195-99.

б) Опишите принципы сертификации программных средств, дайте сравнительную характеристику с сертификацией других видов изделий.

1. а) На основании положений СТБ ИСО/МЭК 12207-2003 дайте сравнительную характеристику функций основных сторон, участвующих в жизненном цикл программных средств и систем.

б) Выполните сравнительный анализ методов оценки качества программных средств, регламентированных в ГОСТ 28195–99 и в ISO/IEC 14598-1:1999.

1. а) На основании положений СТБ ИСО/МЭК 12207-2003 выполните сравнительный анализ различных категорий оценок программных продуктов и работ, проводимых в течение жизненного цикла.

б) Опишите общие черты и различия иерархических моделей качества программных средств, определенных в ISO/IEC 9126-1:2001 и СТБ ИСО/МЭК 9126–2003.

1. а) На основании положений СТБ ИСО/МЭК 12207-2003 охарактеризуйте использование процесса документирования в каждой из работ процесса разработки.

б) На основании положений ISO/IEC 9126-1:2001 перечислите виды метрик качества программных средств и дайте их сравнительную характеристику. Приведите примеры использования каждого из видов метрик.

1. а) На основании положений СТБ ИСО/МЭК 12207-2003 охарактеризуйте использование процесса документирования в каждом из вспомогательных процессов жизненного цикла.

б) Назовите формы подтверждения соответствия, опишите их общие черты и различия.

1. а) На основании положений СТБ ИСО/МЭК 12207-2003 охарактеризуйте использование процесса документирования в каждом из организационных процессов жизненного цикла.

б) Опишите общие черты и различия иерархических моделей сопровождаемости программных средств, определенных в ГОСТ 28195–99 и ISO/IEC 9126-1:2001.

1. а) На основании положений СТБ ИСО/МЭК 12207-2003 охарактеризуйте использование процесса документирования в каждой из работ процессов заказа и поставки.

б) Выполните сравнительный анализ методов оценки качества программных средств, регламентированных в ГОСТ 28195–99 и в ISO/IEC 14598-1:1999.

1. а) На основании положений СТБ ИСО/МЭК 12207-2003 охарактеризуйте возможности использования процесса верификации в процессе заказа.

б) Выполните сравнительный анализ методов оценки качества программных средств, регламентированных в СТБ ИСО/МЭК 9126–2003 и в ISO/IEC 14598-1:1999.

1. а) На основании положений СТБ ИСО/МЭК 12207-2003 охарактеризуйте возможности использования процесса верификации в процессе поставки.

б) Опишите общие черты и различия иерархических моделей надежности программных средств, определенных в ГОСТ 28195–99 и ISO/IEC 9126-1:2001.

1. а) На основании положений СТБ ИСО/МЭК 12207-2003 охарактеризуйте возможности использования процесса верификации в процессе разработки.

б) Перечислите виды документов об оценке соответствия, дайте сравнительную характеристику данных документов.

1. а) На основании положений СТБ ИСО/МЭК 12207-2003 охарактеризуйте возможности использования процесса верификации в жизненном цикле программных средств и систем.

б) Перечислите виды технических нормативных правовых актов, определенных в законе РБ № 262-З «О техническом нормировании и стандартизации», и дайте их сравнительную характеристику.

1. а) На основании положений СТБ ИСО/МЭК 12207-2003 охарактеризуйте возможности использования процесса аттестации в жизненном цикле программных средств и систем.

б) Опишите общие черты и различия иерархической модели удобства использования программных средств, определенной в ГОСТ 28195–99, и иерархической модели практичности программных средств, определенной в ISO/IEC 9126-1:2001.

1. а) Опишите возможности использования инструментальных средств автоматизации жизненного цикла программных средств и систем в работах и задачах процесса разработки.

б) Опишите общие черты и различия иерархических моделей качества программных средств, определенных в ISO/IEC 9126-1:2001 и СТБ ИСО/МЭК 9126–2003.

1. а) Опишите возможности использования инструментальных средств автоматизации жизненного цикла программных средств и систем в работах и задачах процессов заказа и поставки.

б) Опишите общие черты и различия иерархической модели универсальности программных средств, определенной в ГОСТ 28195–99, и иерархической модели мобильности программных средств, определенной в ISO/IEC 9126-1:2001.

1. а) Опишите возможности использования инструментальных средств автоматизации жизненного цикла программных средств и систем во вспомогательных процессах жизненного цикла программных средств и систем.

б) Выполните сравнительный анализ методов оценки качества программных средств, регламентированных в СТБ ИСО/МЭК 9126–2003 и в ГОСТ 28195–99.

### Контрольная работа № 2

Тема задания

Оценка качества и метрология программных средств. Для заданного преподавателем варианта разработать метрику качества, определить критерии обоснованности и свойства разработанной метрики.

Методические указания к выполнению контрольной работы

К выполнению контрольной работы № 2 следует приступать после изучения следующих тем теоретического раздела: тема 5 (Стандартизация качества программных средств за рубежом), 6 (Метрология качества программных средств). В разделе 6 приведены свойства и критерии обоснованности метрик качества программных средств, а также по одному примеру метрики качества на каждую подхарактеристику качества.

В контрольной работе следует, исходя из определения (физического смысла) соответствующей характеристики и подхарактеристики качества, а также определения внутренних, внешних метрик и метрик качества в использовании предложить свою метрику, которая может служить одной из метрик заданной подхарактеристики.

**Контрольную работу следует выполнить и выслать не позднее, чем за две неделю до очередной экзаменационной сессии.**

Содержание отчета по контрольной работе № 2

Контрольная работа №2 должна содержать:

* титульный лист образец титульного листа приведен ниже);
* номер и задание заданного варианта;
* теоретическое описание характеристики и подхарактеристики качества, к которой относится разработанная метрика;
* разработанную метрику качества, представленную в абсолютных единицах;
* разработанную метрику качества, полученную в результате нормирования (представленную в относительных единицах в диапазоне 0..1);
* определение критериев обоснованности и свойств разработанной метрики.

Указания к выбору варианта контрольной работы № 2

Номер варианта индивидуального задания определяется следующим образом: 1 плюс остаток от целочисленного деления номера вашей зачетной книжки (его двух последних цифр, стоящих после дефиса) на сорок шесть.

**Например**, номер варианта студента, имеющего номер зачетки 801021-25, равен 1 + 25 mod 46 = 26; номер варианта студента, имеющего номер зачетки 801021-48, равен: 1 + 48 mod 46 = 3; номер варианта студента, имеющего номер зачетки 801021-46, равен: 1 + 46 mod 46 = 1.

Варианты индивидуальных заданий по контрольной работе № 2

1. Разработать внутреннюю метрику подхарактеристики *Пригодность* ПС.
2. Разработать внешнюю метрику подхарактеристики *Пригодность* ПС.
3. Разработать внутреннюю метрику подхарактеристики *Правильность* ПС.
4. Разработать внешнюю метрику подхарактеристики *Правильность* ПС.
5. Разработать внутреннюю метрику подхарактеристики *Способность к взаимодействию* ПС.
6. Разработать внешнюю метрику подхарактеристики *Способность к взаимодействию* ПС.
7. Разработать внутреннюю метрику подхарактеристики *Защищенность* ПС.
8. Разработать внешнюю метрику подхарактеристики *Защищенность* ПС.
9. Разработать внутреннюю метрику подхарактеристики *Завершенность* ПС.
10. Разработать внешнюю метрику подхарактеристики *Завершенность* ПС.
11. Разработать метрику качества в использовании подхарактеристики *Результативность* ПС.
12. Разработать внутреннюю метрику подхарактеристики *Устойчивость к ошибке* ПС.
13. Разработать внешнюю метрику подхарактеристики *Устойчивость к ошибке* ПС.
14. Разработать внутреннюю метрику подхарактеристики *Восстанавливаемость* ПС.
15. Разработать внешнюю метрику подхарактеристики *Восстанавливаемость* ПС.
16. Разработать внутреннюю метрику подхарактеристики *Понятность* ПС.
17. Разработать внешнюю метрику подхарактеристики *Понятность* ПС.
18. Разработать внутреннюю метрику подхарактеристики *Обучаемость* ПС.
19. Разработать внешнюю метрику подхарактеристики *Обучаемость* ПС.
20. Разработать внутреннюю метрику подхарактеристики *Простота использования* ПС.
21. Разработать внешнюю метрику подхарактеристики *Простота использования* ПС.
22. Разработать метрику качества в использовании подхарактеристики *Продуктивность* ПС.
23. Разработать внутреннюю метрику подхарактеристики *Привлекательность* ПС.
24. Разработать внешнюю метрику подхарактеристики *Привлекательность* ПС.
25. Разработать внутреннюю метрику подхарактеристики *Поведение во времени* ПС.
26. Разработать внешнюю метрику подхарактеристики *Поведение во времени* ПС.
27. Разработать внутреннюю метрику подхарактеристики *Использование ресурсов* ПС.
28. Разработать внешнюю метрику подхарактеристики *Использование ресурсов* ПС.
29. Разработать внутреннюю метрику подхарактеристики *Анализируемость* ПС.
30. Разработать внешнюю метрику подхарактеристики *Анализируемость* ПС.
31. Разработать внутреннюю метрику подхарактеристики *Изменяемость* ПС.
32. Разработать внешнюю метрику подхарактеристики *Изменяемость* ПС.
33. Разработать метрику качества в использовании подхарактеристики *Безопасность* ПС.
34. Разработать внутреннюю метрику подхарактеристики *Стабильность* ПС.
35. Разработать внешнюю метрику подхарактеристики *Стабильность* ПС.
36. Разработать внутреннюю метрику подхарактеристики *Тестируемость* ПС.
37. Разработать внешнюю метрику подхарактеристики *Тестируемость* ПС.
38. Разработать внутреннюю метрику подхарактеристики *Адаптируемость* ПС.
39. Разработать внешнюю метрику подхарактеристики *Адаптируемость* ПС.
40. Разработать внутреннюю метрику подхарактеристики *Настраиваемость* ПС.
41. Разработать внешнюю метрику подхарактеристики *Настраиваемость* ПС.
42. Разработать внутреннюю метрику подхарактеристики *Совместимость* ПС.
43. Разработать внешнюю метрику подхарактеристики *Совместимость* ПС.
44. Разработать метрику качества в использовании подхарактеристики *Удовлетворенность* ПС.
45. Разработать внутреннюю метрику подхарактеристики *Взаимозаменяемость* ПС.
46. Разработать внешнюю метрику подхарактеристики *Взаимозаменяемость* ПС.

### Правила оформления отчетов по контрольным работам

Отчеты по контрольным работам выполняются в текстовом редакторе Word. Необходимо установить следующие размеры страницы:

Размер страницы – 21 х 29,7 см

Поле слева – 2,5 см

Поле справа – 1,5 см

Поле сверху – 2,5 см

Поле снизу – 2,2 см.

Размер шрифта Times New Roman 14, интервал между буквами обычный, интервал между строчками одинарный, выравнивание по ширине.

Объем каждой из контрольных работ должен составлять 4 - 5 страниц. Файл с контрольной работой № 1 должен иметь имя ФамилияК1.doc, с контрольной работой № 2 – ФамилияК2.doc.

### Образец титульного листа

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра программного обеспечения информационных технологий

Факультет НиДО

Специальность ПОИТ

Контрольная работа № 1

по дисциплине «Метрология, стандартизация и сертификация   
в информатике и радиоэлектронике»

Выполнил студент: Иванов И.И.

группа 801021

Зачетная книжка № 801021-20

Минск 2011

## Индивидуальные практические работы, их характеристика

### Теоретические сведения к индивидуальным практическим работам

Общие сведения о метриках сложности программ

Качество программных средств во многом зависит от сложности их кодов. Например, чем сложнее программа, тем ниже ее надежность и сопровождаемость. Поэтому при оценке качества программ обычно оценивается и их сложность.

Метрики сложности программ принято подразделять на ***три основные группы*** [2]:

* метрики размера программ;
* метрики сложности потока управления программ;
* метрики сложности потока данных программ.

**Метрики размера программ**

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*** [3].

Основу метрик Холстеда составляют *шесть базовых метрик* программы:

* η1 *–* словарь операторов (число уникальных операторов программы);
* η2 *–* словарь операндов (число уникальных операндов программы);
* *N1 –* общее число операторов в программе;
* *N2 –* общее число операндов в программе;
* *f1j –* число вхождений *j*-го оператора, *j* = 1, 2, …, η1;
* *f2i –* число вхождений *i*-го операнда, *i* = 1, 2, …, η2.

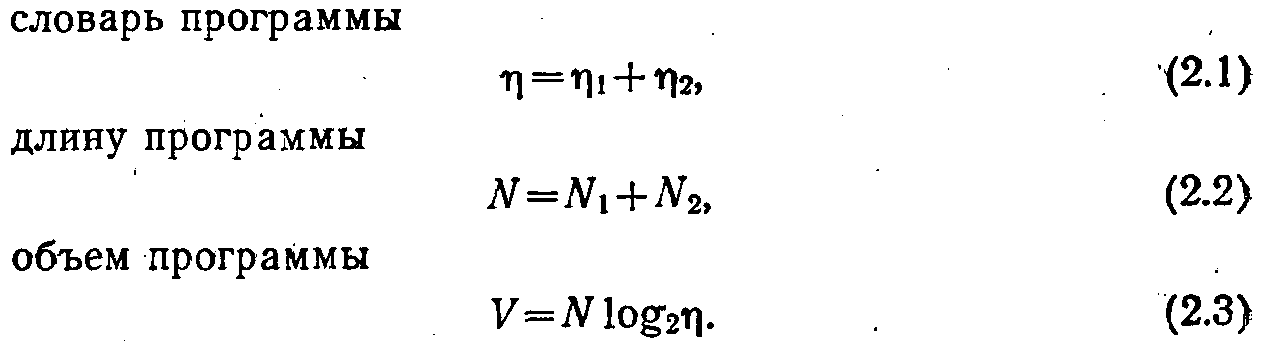
Справедливы следующие соотношения:

****

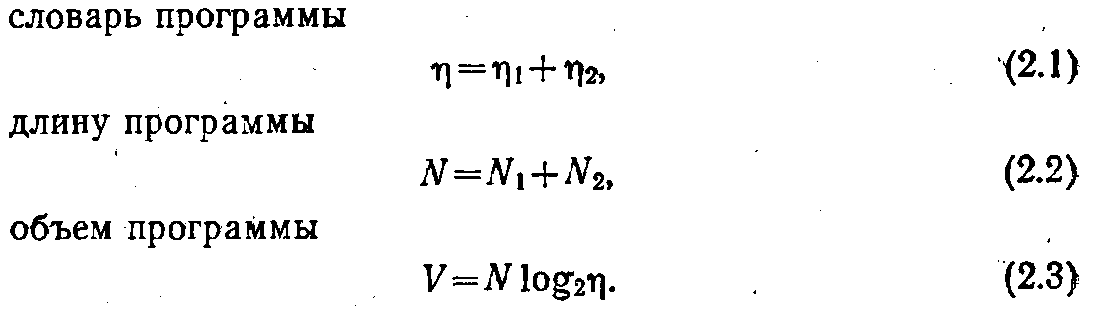
****

Базовые метрики определяются непосредственно при анализе исходных текстов программ. На основе базовых метрик Холстед предложил рассчитывать ряд производных метрик программы. Среди них рассмотрим следующие:

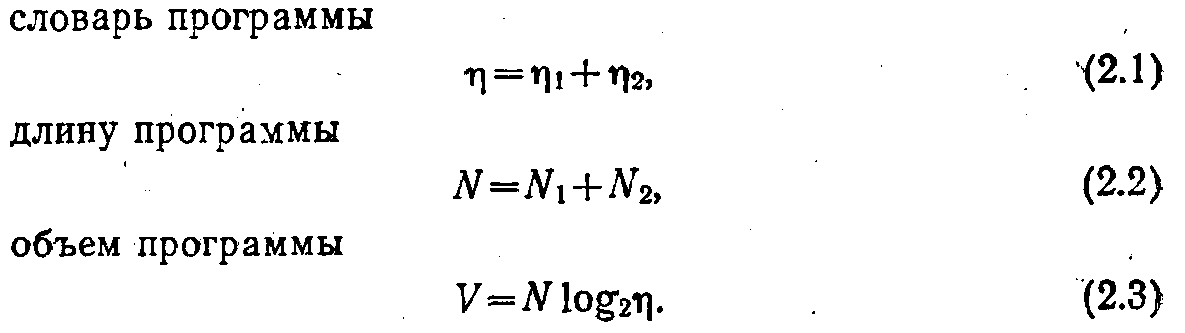
* словарь программы (общее число уникальных операторов и операндов программы):



* длина программы (общее количество операторов и операндов программы):



* объем программы (число битов, т.е. логических единиц информации, необходимых для записи программы):



Операнды программы представляют собой используемые в ней переменные и константы.

Под операторами программы в метриках Холстеда подразумеваются входящие в ее состав символы операций, символ присваивания, символы-разделители точка и точка с запятой, круглая скобка (пара из открывающей и закрывающей скобок считается одним оператором), управляющие операторы, составной оператор, а также имена процедур и функций.

Несколько служебных слов, входящих в состав одного оператора (например, If…Then…Else), считаются одним оператором.

Метки не относятся ни к операторам, ни к операндам.

Очевидно, что совокупность операторов программы и их количество зависят от языка программирования, на котором написана программа.

Операторы языка Паскаль в интерпретации Холстеда приведены в табл. 1. При подсчете количества операторов и операндов в программе, написанной на языке Паскаль, следует анализировать только ее раздел операторов, а также разделы операторов процедур и функций пользователя.

Таблица 1

**Операторы языка Паскаль 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 (знак деления, результат всегда имеет вещественный тип) |
| **<** | меньше |
| **>** | больше |
| **=** | равно |
| **.** | точка (признак конца программы и модуля) |
| **;** | точка с запятой (разделитель операторов программы) |
| **( )** | левая и правая скобки при выделении подвыражений |
| **<=** | меньше или равно |
| **>=** | больше или равно |
| **<>** | не равно |
| **:=** | операция присваивания |
| **^** | знак карата (обращение к динамической переменной) |
| **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 |
| **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 |

**Пример 1.** Расчет метрик Холстеда для программы, вычисляющей значение функции

***Y = sin X***

через разложение функции в бесконечный ряд

f156

с точностью ***Eps = 0,0001***.

Текст программы, написанной на языке Паскаль, 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 приведены в табл. 2.

Таблица 2

**Расчет базовых метрик Холстеда для программы, вычисляющей значение функции *Y = sin X***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***j*** | **Оператор** | ***f1j*** | ***i*** | **Операнд** | ***f2i*** |
|  | ; | 7 |  | x | 6 |
|  | := | 6 |  | n | 5 |
|  | \* | 4 |  | vs | 5 |
|  | – | 3 |  | y | 4 |
|  | / | 2 |  | 2 | 4 |
|  | ( ) | 2 |  | 1 | 2 |
|  | + | 2 |  | eps | 2 |
|  | Begin…End | 1 |  |  |  |
|  | Readln ( ) | 1 |  |  |  |
|  | Repeat…Until | 1 |  |  |  |
|  | abs( ) | 1 |  |  |  |
|  | < | 1 |  |  |  |
|  | Writeln ( ) | 1 |  |  |  |
|  | **.** | 1 |  |  |  |
| **η1 *=* 14** |  | ***N1 =* 33** | **η2 *=* 7** |  | ***N2 =* 28** |

Словарь программы **η**=14 + 7 = 21.

Длина программы ***N*** = 33 + 28 = 61.

Объем программы ***V*** = 

**Метрики сложности потока управления программ**

Метрики сложности потока управления программ принято определять на основе представления программ в виде управляющего ориентированного графа *G=(V, E)*, где *V —* вершины, соответствующие операторам, а *Е —* дуги, соответствующие переходам между операторами. В дуге *(v, и)* вершина *v* является исходной, а *и —* конечной. При этом *и* непосредственно следует за *v,* a *v* непосредственно предшествует *и.* Если путь от *v* до *u* состоит более чем из одной дуги, тогда *и* следует за *v,* а *v* предшествует *и* [2]*.*

Частным случаем представления ориентированного графа программы можно считать детализированную схему алгоритма, в которой каждому блоку соответствует один оператор программы, построенную в соответствии с положениями стандарта ГОСТ 19.701-90 [1]. 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 – ʋ + 2p*,**

где *е —* число дуг ориентированного графа *G; ʋ —* число вершин; *р —* 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 (разветвлений).

Рассчитаем метрику Маккейба для программы, схема алгоритма которой приведена на рис. 1. Действия, выполняемые блоками программы, в примере не показаны. Внутри каждого блока помещены их номера. Компонент связности графа обозначен штриховой дугой. Число дуг *е* = 8, число вершин *ʋ* = 7*,* *р* = 1. Цикломатическое число Маккейба равно *Z(G)* = 8 – 7 +2 = 3.

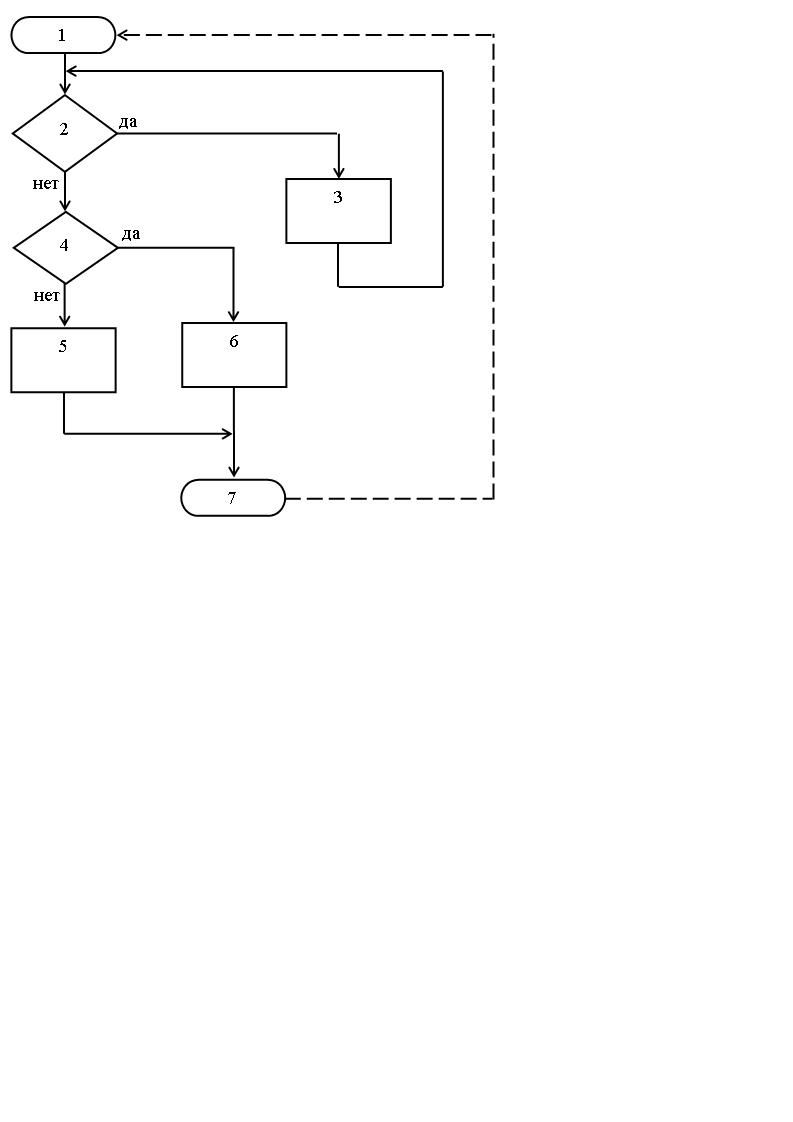


Рис. 1. Пример схемы алгоритма программы

Значение метрики Маккейба показывает, что в схеме алгоритма (см. рис.1) можно выделить три базисных независимых пути (называемых также линейно независимыми контурами):

1-й путь. 1 – 2 (нет) – 4 (нет) – 5 – 7.

2-й путь. 1 – 2 (да) – 3 – 2 (нет) – 4 (нет) – 5 – 7.

3-й путь. 1 – 2 (нет) – 4 (да) – 6 – 7.

Вторым возможным вариантом совокупности базисных независимых путей является:

1-й путь. 1 – 2 (да) – 3 – 2 (нет) – 4 (нет) – 5 – 7.

2-й путь. 1 – 2 (нет) – 4 (нет) – 5 – 7.

3-й путь. 1 – 2 (да) – 3 – 2 (нет) – 4 (да) – 6 – 7.

Таким образом, для тестирования совокупности базисных независимых путей исследуемой программы необходимо выполнить минимально три тестовых прогона.

***Метрика Джилба*** определяет логическую сложность программы как насыщенность программы условными операторами IF–THEN–ELSE. Обычно используются два вида метрики Джилба: *CL —* количество условных операторов, характеризующее абсолютную сложность программы; *cl —* насыщенность программы условными 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 оператора* *CLI*.

Использование в программе оператора выбора (например, CASE) с *n* разветвлениями эквивалентно применению *n* – 1 оператора IF–THEN–ELSE с глубиной вложенности *n* – 2.

Например. на рис. 2 приведена схема алгоритма вычисления некоторой функции Y. В данной схеме используется выбор, обозначаемый символом «Решение» с пятью разветвлениями (*n* = 5). Эквивалентный алгоритм вычисления той же функции Y. использующий несколько операторов IF–THEN–ELSE, представлен на рис. 3. На данном рисунке количество операторов IF–THEN–ELSE равно четырем (*n* – 1), максимальный уровень вложенности оператора IF–THEN–ELSE равен трем (*n* – 2).

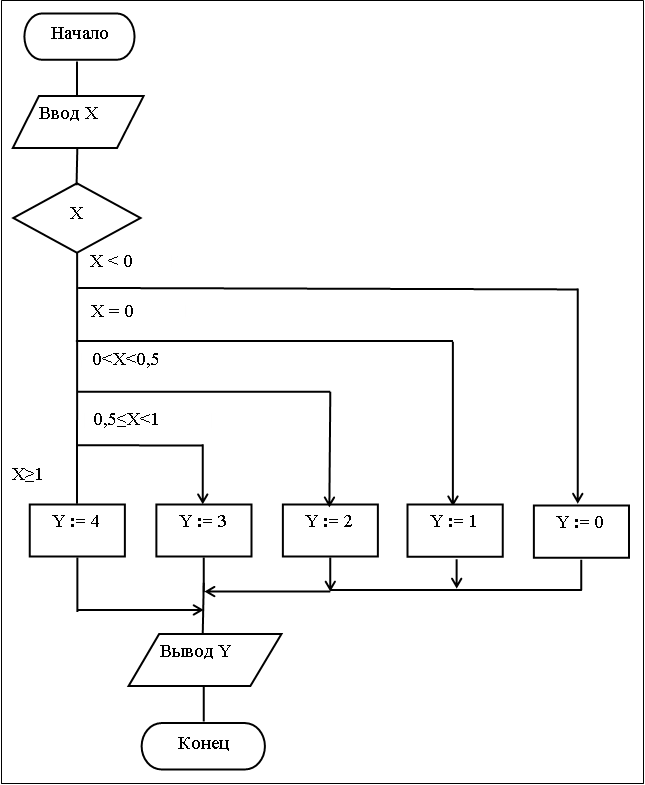


Рис. 2. Схема разветвляющегося алгоритма вычисления функции ***Y***   
(используется символ «Решение» с многими выходами)

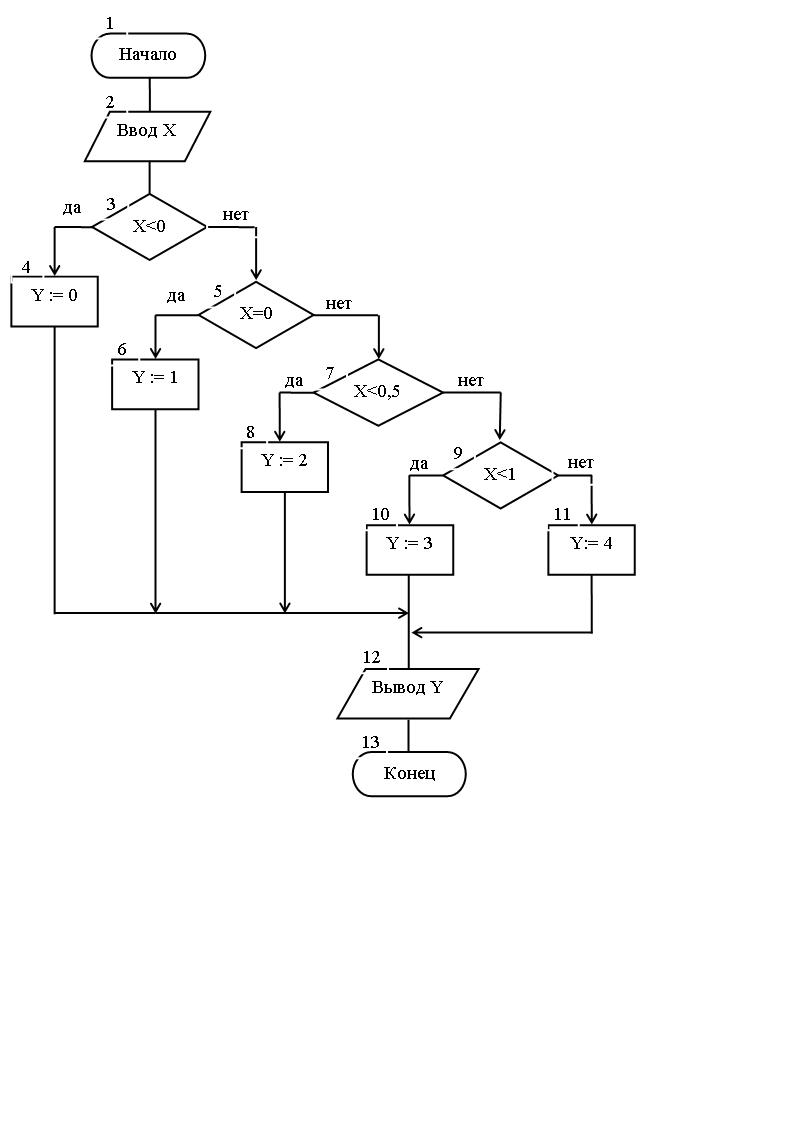


Рис. 3. Схема разветвляющегося алгоритма вычисления функции ***Y***   
(используется символ «Решение» с двумя выходами)

Таким образом, для схем алгоритмов, приведенных на рис. 2 и 3, *CL* = 4, *cl =* 0,36 (количество операторов программы равно 11), *CLI =* 3.

Значения метрики Маккейба для данных алгоритмов также совпадают. Для схемы алгоритма, представленной на рис. 2,

*Z(G)* = 13 – 10 +2 = 5.

Для схемы алгоритма, приведенной на рис. 3,

*Z(G)* = 16 – 13 +2 = 5.

Следует отметить, что сложность программы с помощью метрики Джилба не всегда возможно посчитать на основе схемы алгоритма, так как схема алгоритма может быть представлена укрупнено. 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.

***Метрика граничных значений*** базируется на определении скорректированной сложности вершин графа программы [2].

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.

Число вершин, образующих подграф (исключая вершину выбора, через которую осуществляется вход в подграф), равно скорректированной сложности вершины выбора. Каждая принимающая вершина имеет скорректированную сложность, равную 1. Конечная вершина имеет скорректированную сложность, равную 0.

*Абсолютная граничная сложность программы* *Sa* определяется как сумма скорректированных сложностей всех вершин графа *G*.

*Относительная граничная сложность программы* *So* определяется по формуле:

****

где *So –* относительная граничная сложность программы; *Sa —* абсолютная граничная сложность программы; *ʋ –* общее число вершин графа программы.

В таблице 3 представлены свойства подграфов программы, схема алгоритма которой приведена на рис. 3. Табл. 4 содержит скорректированные сложности вершин графа данной программы. Номера вершин графа соответствуют номерам соответствующих блоков на схеме алгоритма.

Таблица 3

**Свойства подграфов программы (рис. 3)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Свойства подграфов программы | Номер вершины выбора | | | |
| 3 | 5 | 7 | 9 |
| Номера вершин перехода | 4, 5 | 6, 7 | 8, 9 | 10, 11 |
| Скорректированная сложность вершины выбора | 9 | 7 | 5 | 3 |
| 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 |

Таблица 4

**Скорректированные сложности вершин графа программы (рис. 3)**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Номер вершины графа программы |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Скорректированная сложность вершины графа | 1 | 1 | 9 | 1 | 7 | 1 | 5 | 1 | 3 | 1 | 1 | 1 | 0 | ***Sa* = 32** |

Таким образом, абсолютная граничная сложность *Sa* программы, схема алгоритма которой приведена на рис. 3, равна 32. Относительная граничная сложность данной программы равна

*So*= l – (13 – 1)/32 = 0,625.

В табл. 5 представлены свойства подграфов программы, схема алгоритма которой приведена на рис. 1. Табл. 6 содержит скорректированные сложности вершин графа данной программы.

Таблица 5

**Свойства подграфов программы (рис. 1)**

|  |  |  |
| --- | --- | --- |
| Свойства подграфов программы | Номер вершины выбора | |
| 2 | 4 |
| Номера вершин перехода | 3, 4 | 5, 6 |
| Скорректированная сложность вершины выбора | 3 | 3 |
| Номера вершин подграфа | 2, 3 | 5, 6 |
| Номер нижней границы подграфа | 4 | 7 |

Таблица 6

**Скорректированные сложности вершин графа программы (рис. 1)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Номер вершины графа программы |  |  |  |  |  |  |  |  |
| Скорректированная сложность вершины графа | 1 | 3 | 1 | 3 | 1 | 1 | 0 | ***Sa* = 10** |

Абсолютная граничная сложность *Sa* программы, схема алгоритма которой приведена на рис. 1, равна 10. Относительная граничная сложность данной программы равна

*So*= l – (7 – 1)/10 = 0,4.

Табл. 7 содержит метрики сложности потока управления для программ, схемы алгоритмов которых приведены на рис. 1 – 3.

Таблица 7

**Метрики сложности потока управления программ**

|  |  |  |  |
| --- | --- | --- | --- |
| Метрики сложности потока управления | Схемы алгоритмов | | |
| Рис. 1 | Рис. 2 | Рис. 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 | 0,29 \* | 0,36 | 0,36 |
| Максимальный уровень вложенности условного оператора *CLI* по метрике Джилба | 1 | 3 | 3 |
| Метрика граничных значений (абсолютная граничная сложность программы) *Sa* | 10 | 14 | 32 |
| Метрика граничных значений (относительная граничная сложность программы) *S0* | 0,4 | 0,357 | 0,625 |

Примечание.

\* - относительная сложность *cl* по метрике Джилба для программы, схема алгоритма которой представлена на рис. 1, рассчитана в предположении, что каждый блок схемы алгоритма содержит один оператор программы.

**Метрики сложности потока данных**

Метрики сложности потока данных связывают сложность программы с использованием и размещением данных в данной программе. Метрики этой группы основаны на анализе исходных текстов программ.

К наиболее известным в данной группе метрик можно отнести спен и метрику Чепина [2].

***Спен идентификатора*** – это число повторных появлений идентификатора (число появлений после его первого появления) в тексте программы. Идентификатор, встречающийся в тексте программы *п* раз, имеет спен, равный *п—*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 в программе переменных.

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. *Р –* вводимые переменные, содержащие исходную информацию, но не модифицируемые в программе и не являющиеся управляющими переменными;

2. *М –* модифицируемые переменные и создаваемые внутри программы константы и переменные, не являющиеся управляющими переменными;

3. *С –* переменные, участвующие в управлении работой программного модуля (управляющие переменные);

4. *Т –* 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 = a1 p + a2 m + a3 c + a4 t ,*

где *а1, a2,* *a3, a4* – весовые коэффициенты; *p, m, c, t* – количество переменных в группах *Р, М, С, Т* соответственно.

Весовые коэффициенты позволяют учитывать различное влияние на сложность программы каждой функциональной группы. Наиболее часто применяются следующие значения весовых коэффициентов: *а1* = l, *а2* = 2, *а3* = 3, *а4* = 0,5. С учетом данных значений формула для определения метрики Чепина принимает вид :

*Q = p* + 2*m* + 3*c* + 0,5*t .*

Помимо полной метрики Чепина распространен ее вариант, при котором анализу и разбиению на четыре группы подвергаются только переменные из списка ввода/вывода программы, то есть те переменные, которые содержатся в списке параметров операторов ввода/вывода программы. Назовем данный вариант метрикой Чепина ввода/вывода.

Табл. 8, 9 содержат метрики сложности потока данных для программы, вычисляющей значение функции

***Y = sin X .***

Исходный текст программы приведен в примере 1.

Таблица 8

**Спен программы**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Идентификатор | х | n | vs | y | eps | Суммарный спен программы |
| Спен | 6 | 5 | 5 | 4 | 2 | 22 |

В тексте программы (см. пример 1) идентификаторы впервые встречаются в разделе объявлений. Поэтому значение спена идентификатора равно количеству его появлений в разделе операторов, то есть значению *f2i* в метриках Холстеда.

Таблица 9

**Метрики Чепина программы**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Полная метрика Чепина | | | | Метрика Чепина ввода/вывода | | | |
| Группа переменных | Р | М | С | Т | Р | М | С | Т |
| Переменные, относящиеся к группе | x | y, n | vs, eps | -- | x | y | eps | -- |
| Количество переменных в группе | р = 1 | m = 2 | c = 2 | t = 0 | р = 1 | m = 1 | c = 1 | t = 0 |
| Метрика Чепина | Q = 1\*1 + 2\*2 + 3\*2 + 0,5\*0 = 11 | | | | Q = 1\*1 + 2\*1 + 3\*1 + 0,5\*0 = 6 | | | |

В список переменных ввода/вывода программы (см. пример 1) входят переменные *х* (оператор ввода *Readln (x)*) и *y, z* (оператор вывода *Writeln (x, y, eps)*). Остальные переменные (*n, vs*) в расчете метрики Чепина ввода/вывода не участвуют.

**Литература**

1. ГОСТ 19.701-90 – Единая система программной документации – Схемы алгоритмов, программ, данных и систем – Условные обозначения и правила выполнения.
2. Изосимов А. В. Метрическая оценка качества программ / А. В. Изосимов, Рыжко А. Л. – М.: Изд-во МАИ, 1989.
3. Холстед М.Х. Начала науки о программах. – М.: Финансы и статистика, 1981.

### Указания к выбору варианта индивидуального задания для индивидуальных практических работ № 1 и № 2

Номер варианта индивидуального задания студента для лабораторных работ № 1 и № 2 определяется *по правилу*:

1 плюс остаток от *целочисленного* деления номера зачетной книжки студента (двух последних цифр, стоящих после дефиса) на тридцать.

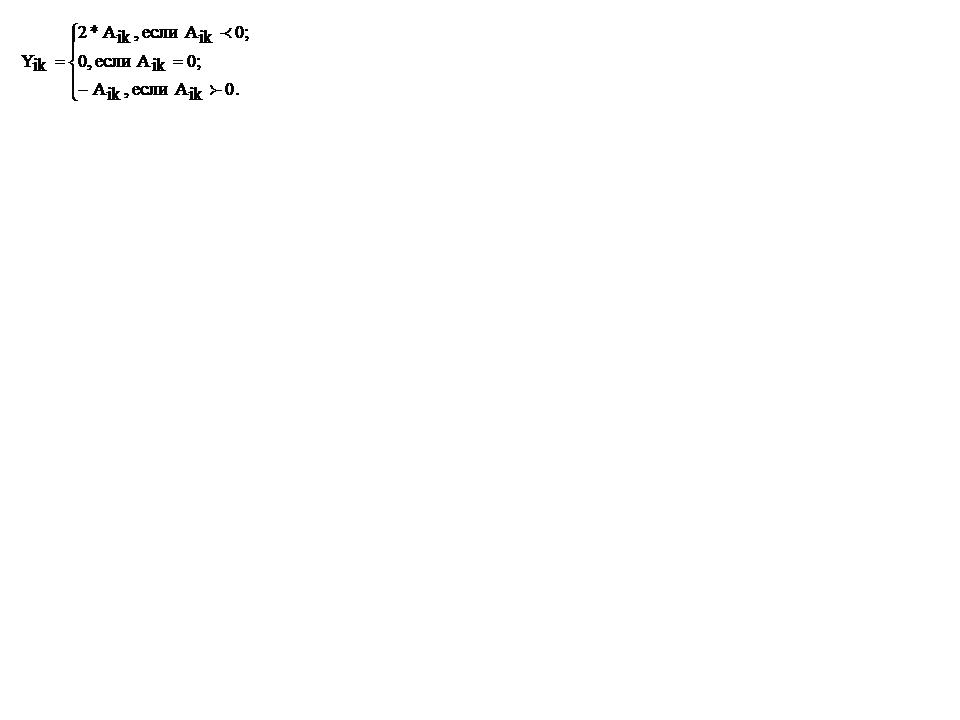
**Например**, номер варианта студента, имеющего номер зачетки 801021-12, равен: 1 + 12 mod 30 = 13.

Номер варианта для студента, имеющего зачетку с номером 801021-35, равен: 1 + 35 mod 30 = 6.

**Индивидуальные практические работы следует выполнить и выслать не позднее, чем за неделю до очередной экзаменационной сессии.**

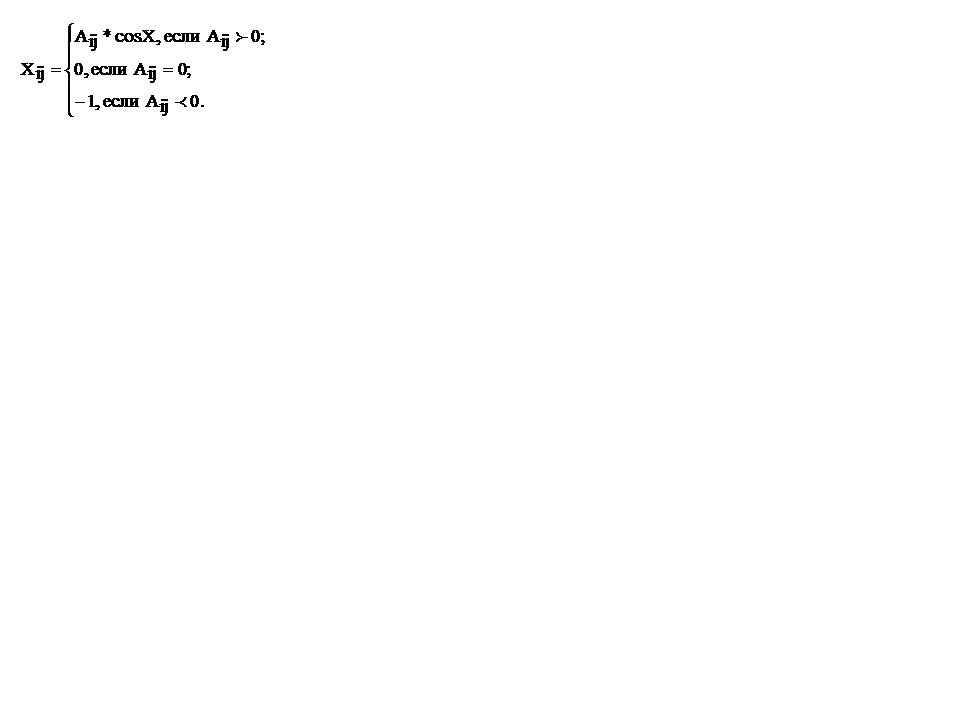
### Варианты индивидуальных заданий для индивидуальных практических работ № 1 и № 2

1. Из последовательности чисел А1, А2, ..., А30 выбрать отрицательные четные числа. Их значения поместить в массив В (30). Остаток массива В заполнить нулями. Вывести исходные числа и массив В.
2. Ввести массив А (10, 10). Вычислить элементы массива Y по формуле



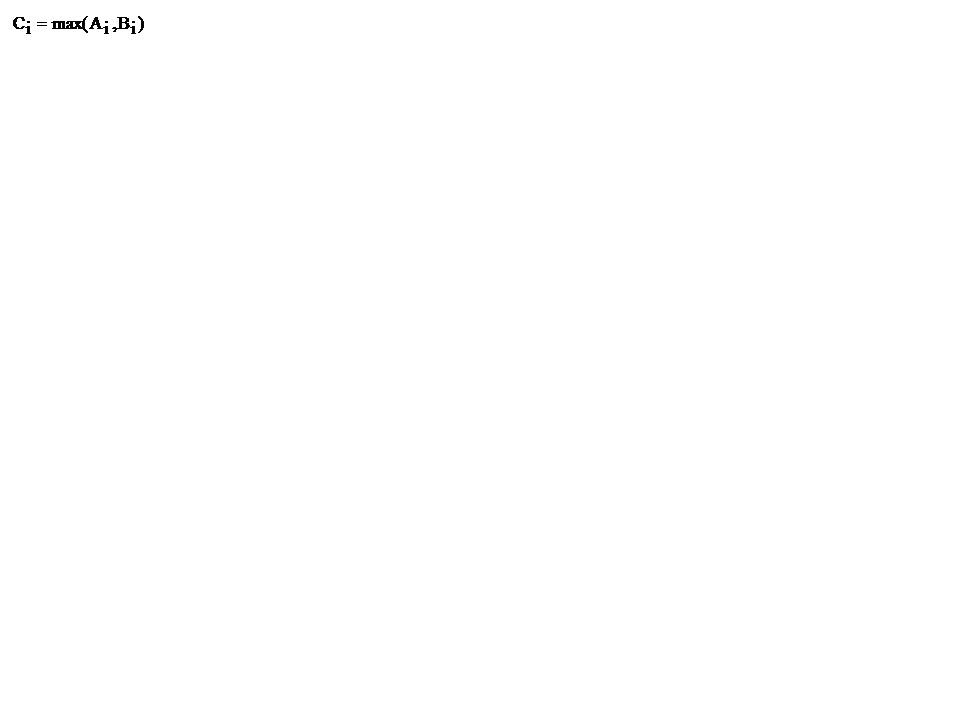
Вывести исходный и результирующий массивы и количество элементов массива Y, получивших значение 0.

1. Ввести массив А (5, 7). Найти сумму элементов каждой строки, максимальную и минимальную из этих сумм. Вывести массив, полученные суммы, номера строк, где находятся максимальная и минимальная суммы.
2. Ввести массив А (6, 7). Вычислить элементы массива Х по формуле



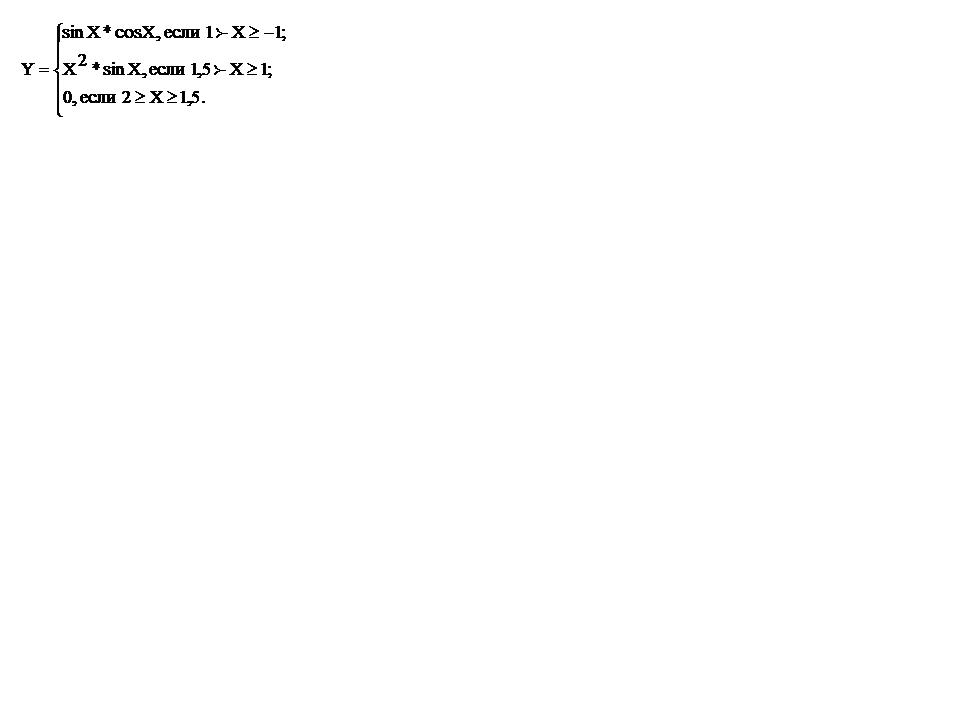
Вывести исходный и результирующий массивы и количество элементов массива Y, получивших значение –1.

1. Ввести массивы А (20) и В (20). Образовать массив С (20) из элементов, встречающихся в обоих массивах. Остаток массива В заполнить нулями. Вывести исходные и результирующий массивы.
2. Ввести массивы А (8) и В (8). Получить массив С (8), элементы которого получают значения по правилу

,

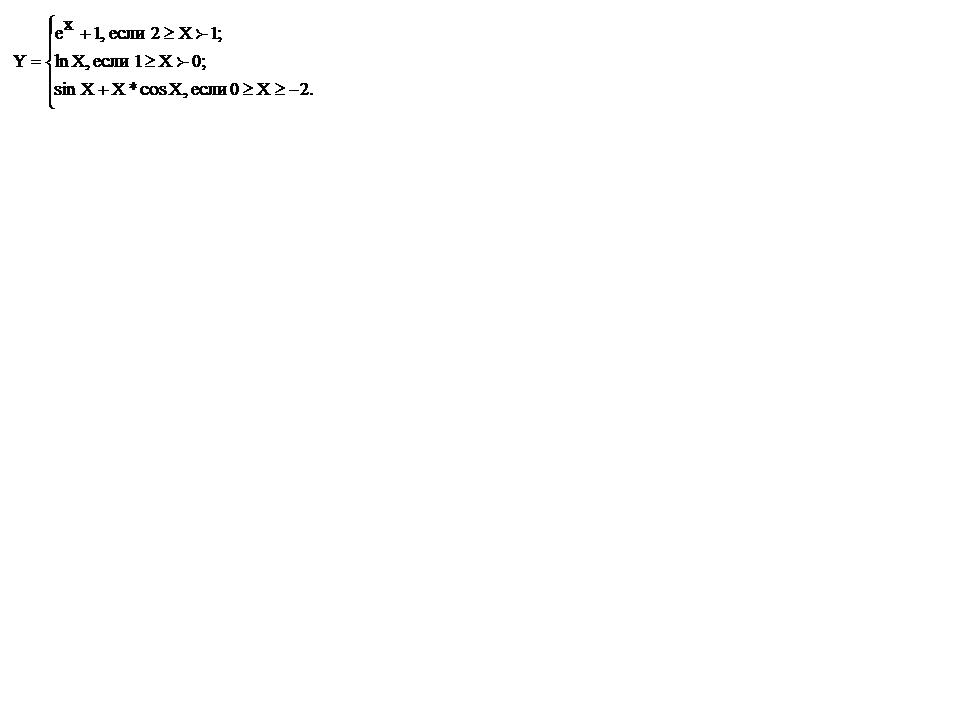
и подсчитать, сколько элементов Сi получило значение Ai. Вывести исходные массивы и результаты вычислений.

1. Ввести массив А (10), первые девять элементов которого упорядочены по возрастанию. Поместить последний элемент массива в соответствующее место массива, чтобы не нарушить его упорядоченность. Вывести исходный и результирующий массивы и номер помещенного элемента.
2. Для значений Х, изменяющихся от –1 до 2 с шагом 0,1 , вычислить значения функции Y:



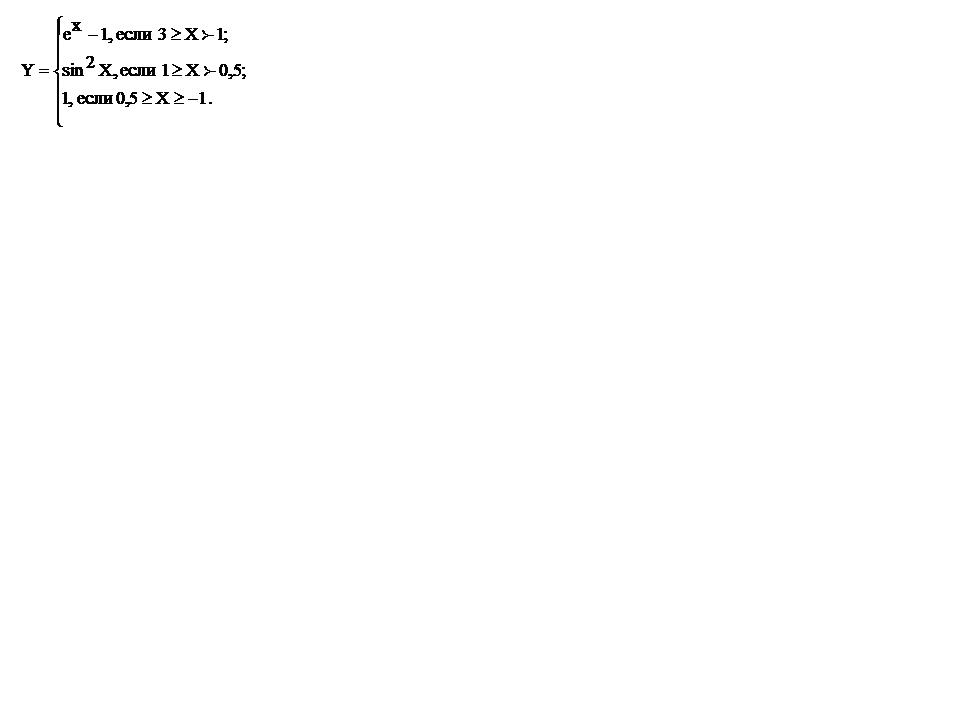
Полученные значения Y занести в массив. Вывести значения Х и соответствующие им значения элементов результирующего массива.

1. Ввести массив из 40 элементов. Посчитать количество положительных элементов в массиве. Сформировать массив, у которого первым элементом будет последний (40) элемент исходного массива. Вторым – предпоследний (39) элемент и т.д. (т.е. расположить элементы в обратном порядке). Новый массив сформировать на месте исходного. Вспомогательный массив не использовать. Вывести исходный и результирующий массивы и количество положительных элементов.
2. Ввести массив А (10, 10). Найти максимальный элемент в главной диагонали и минимальный элемент в побочной диагонали массива А и поменять данные элементы местами. Вывести максимальный и минимальный элементы, номера строк и столбцов, в которых они находятся, исходный и результирующий массивы.
3. Для значений Х, изменяющихся от –2 до 2 с шагом 0,2 , вычислить значения функции Y:



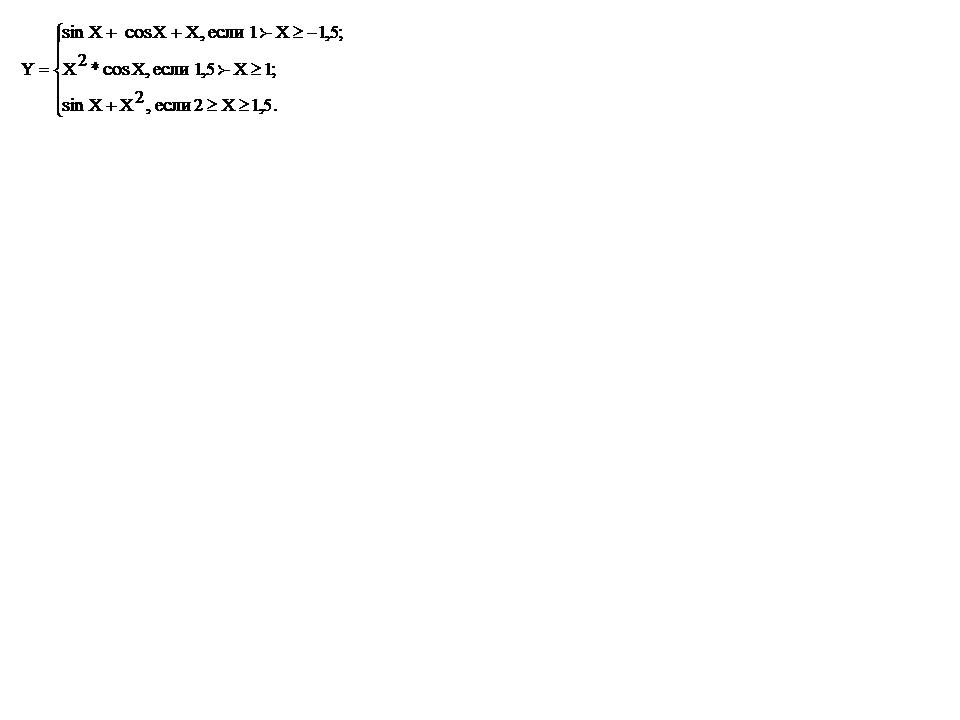
Полученные значения Y занести в массив. Вывести значения Х и соответствующие им значения элементов результирующего массива.

1. В массиве из 100 элементов найти первые десять элементов, которые больше 15. Вывести их значения и их номера в исходном массиве. Если их меньше десяти, вывести текст “Таких значений больше нет”. Если их нет вообще, вывести текст “Таких значений нет”.
2. Ввести массив А (8, 8). Если минимальный элемент данного массива отрицателен, поменять местами главную и побочную диагонали массива. Вывести минимальный элемент, номера строки и столбца, в которых он находится, исходный и результирующий массивы.
3. Для значений Х, изменяющихся от –1 до 3 с шагом 0,25 , вычислить значения функции Y:



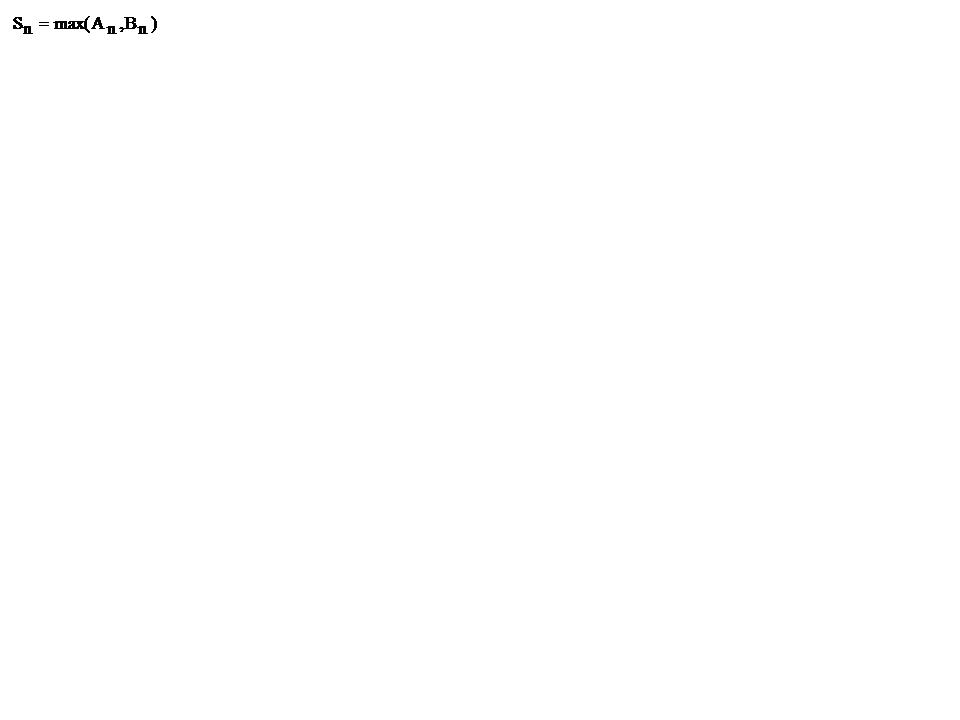
Полученные значения Y занести в массив. Вывести значения Х и соответствующие им значения элементов результирующего массива.

1. В матрице А (10, 10) найти максимальные элементы в строках и максимальный элемент матрицы. Вывести исходную матрицу, найденные значения элементов и номера строк и столбцов, где они находятся.
2. Ввести массив А (8, 8). Найти минимальный элемент в побочной диагонали массива А. Поменять местами столбец, в котором находится данный элемент, с первым столбцом массива А. Вывести максимальный элемент, номера строки и столбца, в которых он находится, исходный и результирующий массивы.
3. Для значений Х, изменяющихся от –1,5 до 2 с шагом 0,01, вычислить значения функции Y:



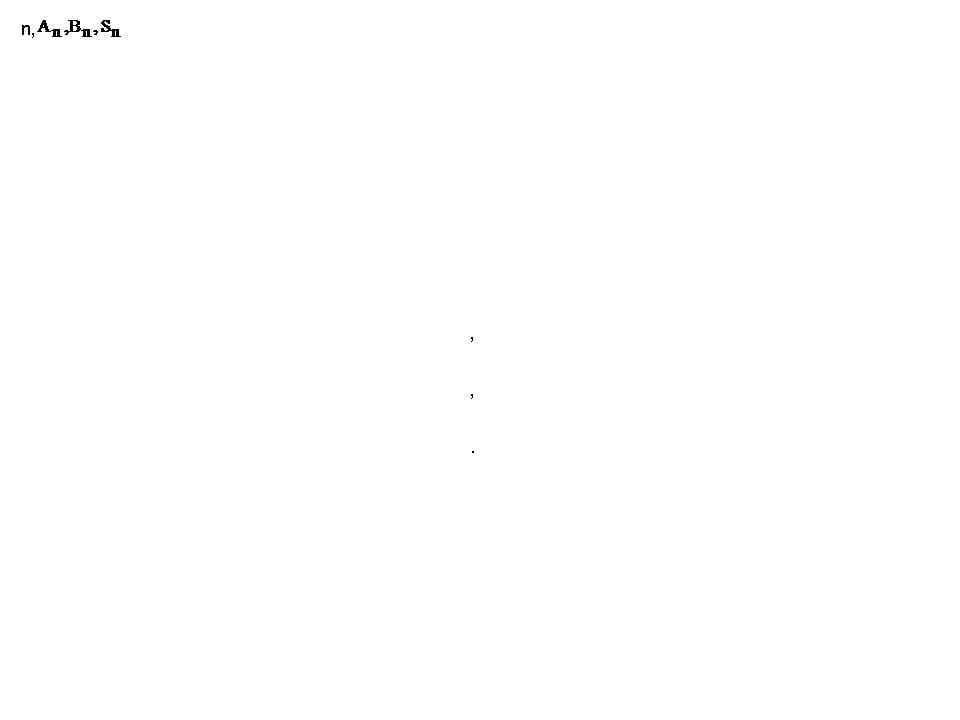
Полученные значения Y занести в массив. Вывести значения Х и соответствующие им значения элементов результирующего массива.

1. Ввести массив А (10, 10). Заменить элементы массива, находящиеся в строках и столбцах с четными номерами, максимальным элементом данного массива. Вывести максимальный элемент, номера строки и столбца, в которых он находится, исходный и результирующий массивы.
2. Вычислить элементы массива S(10) по формуле:

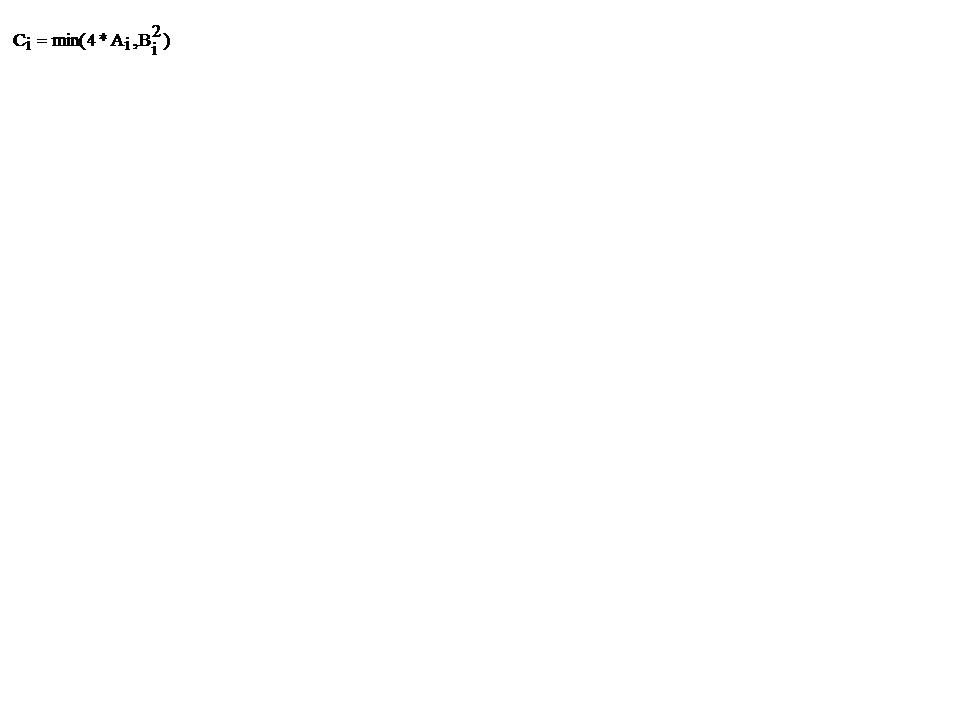


если



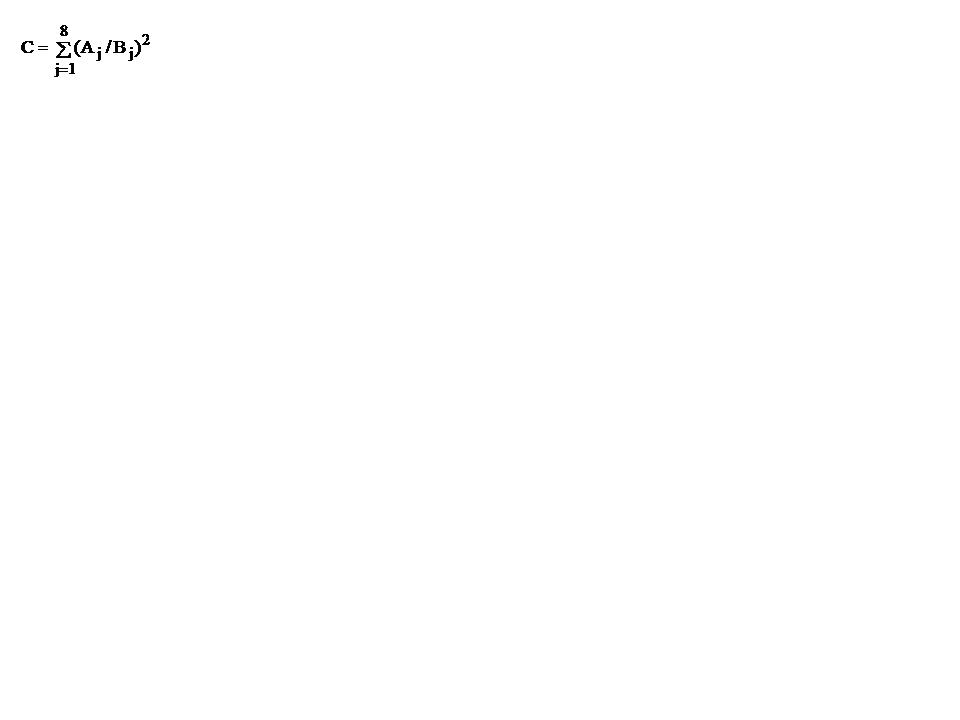
Вывести значения .

1. Ввести массив А (7, 5). Сформировать одномерный массив В (35) из четных положительных элементов массива А. Остаток массива В заполнить нулями. Вывести массивы А и В.
2. Ввести массивы А (8) и В (8). Получить массив С (8), элементы которого формируются по правилу



и подсчитать, сколько элементов  получило значение . Вывести значения массивов А, В, С и полученное количество элементов.

1. Ввести массивы А (4, 5) и В (5, 7). Поменять местами строку массива А, содержащую максимальный элемент данного массива, и столбец массива В, содержащий минимальный элемент массива В. Вывести максимальный и минимальный элементы, номера строк и столбцов, в которых они находятся, исходные и результирующие массивы.
2. Ввести массивы А (8) и В (8). Вычислить



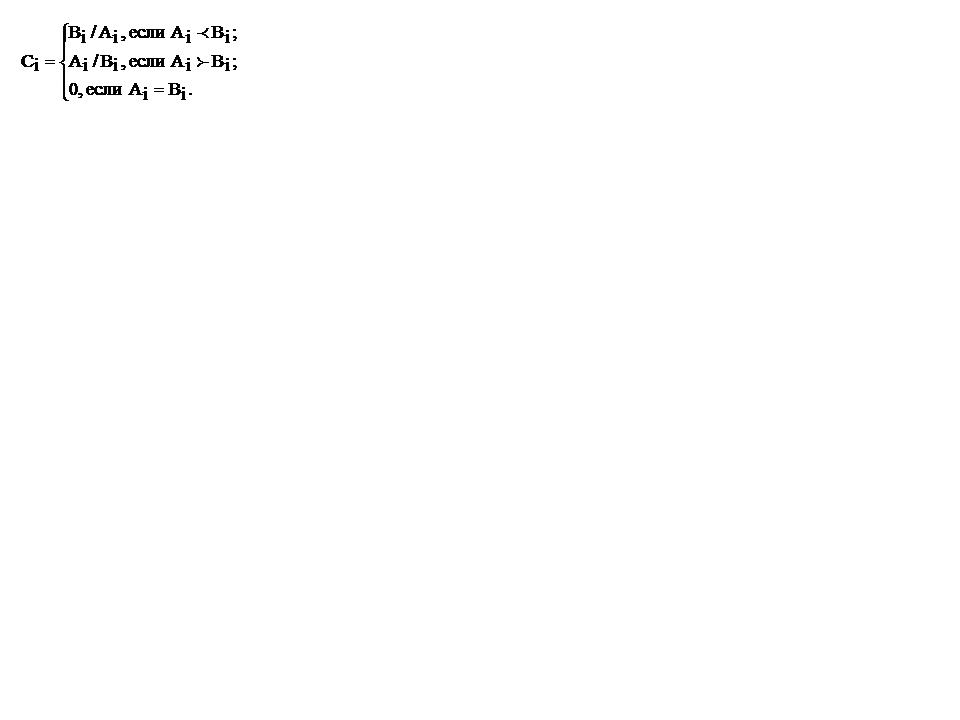
для пар Aj и Bj , удовлетворяющих условию Aj > Bj . Вывести А, В, С и номера элементов массивов, участвующих в вычислениях С.

1. Ввести массив А (7, 7). Найти максимальный и минимальный элементы в побочной диагонали и поменять местами столбцы массива, в которых они находятся. Вывести максимальный и минимальный элементы, номера столбцов, в которых они находятся, исходный и результирующий массивы.
2. Ввести массивы X (6) и Y (6). В массиве X заменить значения тех элементов Xi , для которых выполняется условие



значениями элементов Yi. Вывести исходные и результирующий массивы.

1. Ввести массивы А (5, 7) и В (3, 6). Если максимальный элемент массива А больше минимального элемента массива В, поменять данные элементы местами. Вывести максимальный и минимальный элементы, номера строк и столбцов, в которых они находятся, исходные и результирующие массивы.
2. В массиве А (7, 8) найти минимальные элементы в столбцах и минимальный элемент массива. Вывести исходный массив, найденные значения минимальных элементов и номера строк и столбцов, где они находятся.
3. Ввести массивы А (10) и В (10). Получить массив С (10), элементы которого получают значения по правилу



и подсчитать, сколько элементов массива С получило значение 0. Вывести исходные массивы и результаты вычислений.

1. Ввести массив А (6, 6). Найти максимальный и минимальный элементы в главной диагонали и поменять местами строки массива, в которых они находятся. Вывести максимальный и минимальный элементы, номера строк и столбцов, в которых они находятся, исходный и результирующий массивы.
2. Ввести массив А (7, 8). Найти сумму элементов каждого столбца, максимальную и минимальную из этих сумм. Вывести массив, полученные суммы, номера столбцов, где находятся максимальная и минимальная суммы.

### Правила оформления отчета по индивидуальной практической работе

Отчеты по работам выполняются в текстовом редакторе Word. Необходимо установить следующие размеры страницы:

Размер страницы – 21 х 29,7 см

Поле слева – 2,5 см

Поле справа – 1,5 см

Поле сверху – 2,5 см

Поле снизу – 2,2 см.

Размер шрифта Times New Roman 14, интервал между буквами обычный, интервал между строчками одинарный, выравнивание по ширине.

Объем каждой из индивидуальных практических работ должен составлять 5 - 7 страниц. Файл с индивидуальной практической работой № 1 должен иметь имя ФамилияИР1.doc, с индивидуальной практической работой № 2 – ФамилияИР2.doc.

### Образец титульного листа

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра программного обеспечения информационных технологий

Факультет НиДО

Специальность ПОИТ

Индивидуальная практическая работа № 1

по дисциплине «Метрология, стандартизация и сертификация в информатике и радиоэлектронике»

Вариант № 11

Выполнил студент: Иванов И.И.

группа 801021

Зачетная книжка № 801021-40

Минск 2011

### Индивидуальная практическая работа № 1

Тема задания

***Метрики сложности потока управления программ****.*

Для вычисленного варианта индивидуального задания разработать детализированную схему алгоритма, представленную в соответствии с положениями ГОСТ 19701-90. По полученному алгоритму рассчитать метрики сложности потока управления программ (метрики Маккейба, Джилба, максимальный уровень вложенности условного оператора, метрику граничных значений).

В алгоритме предусмотреть вывод на экран всех входных и выходных данных.

Методические указания к выполнению индивидуальной практической работы

Перед началом выполнения работы из теоретических сведений к индивидуальным практическим работам необходимо изучить материал, касающийся метрик сложности потока управления программ.

Затем для рассчитанного варианта индивидуального задания следует разработать схему алгоритма с максимальным уровнем детализации. Это означает, что каждому блоку схемы алгоритма должен соответствовать один оператор языка программирования (в привязке к языку Паскаль). Например, ввод или вывод двухмерного массива в алгоритме должен быть представлен не одним символом ввода (параллелограмм), а сложным циклом (во внешнем цикле вводятся строки, во внутреннем цикле вводятся элементы строк).

Схема алгоритма должна быть описана. В описании должно быть приведено назначение входных, выходных и внутренних переменных, назначение блоков алгоритма.

На основании разработанного алгоритма рассчитываются значения метрик сложности потока управления будущей программы:

* рассчитывается метрика Маккейба и определяются базисные независимые пути в алгоритме (по аналогии с примером, приведенным в теоретических сведениях);
* рассчитываются абсолютная *CL* и относительная *cl* сложности программы, а также максимальный уровень вложенности условного оператора *CLI,* используя метрику Джилба;
* рассчитываются абсолютная *Sa* и относительная *So* граничные сложности программы по метрике граничных значений (по аналогии с примерами, приведенными в теоретических сведениях). Результаты расчетов метрики граничных значений должны быть представлены в виде таблиц, аналогичных таблицам 3 и 4;
* значения всех рассчитанных метрик сложности потока управления для разработанного алгоритма должны быть сведены в итоговую таблицу (аналогичную таблице 7, но включающую не три столбца для схем алгоритмов, а один столбец для разработанного алгоритма).

Содержание отчета по индивидуальной практической работе № 1

Индивидуальная практическая работа № 1 должна содержать:

* титульный лист (образец титульного листа приведен выше);
* номер и условие индивидуального задания;
* детализированную схему алгоритма по ГОСТ 19. 701-90;
* описание схемы алгоритма;
* расчет метрики Маккейба ддя разработанного алгоритма и определение базисных независимых путей;
* расчет метрики Джилба ддя разработанного алгоритма;
* расчет метрики граничных значений ддя разработанного алгоритма с результатами, представленными в виде таблиц;
* результаты расчетов метрик в виде итоговой таблицы.

### Индивидуальная практическая работа № 2

Тема задания

***Метрики Холстеда. Метрики сложности потока данных.***

Для разработанной в первой индивидуальной практической работе схемы алгоритма разработать текст программы на языке Паскаль. По тексту программы рассчитать метрики Холстеда и метрики сложности потока данных (спен и метрику Чепина).

Методические указания к выполнению индивидуальной практической работы

Перед началом выполнения работы из теоретических сведений к индивидуальным практическим работам необходимо изучить материал, касающийся метрик Холстеда и метрик сложности потока данных.

Затем для разработанной (в соответствии с индивидуальным заданием) в первой индивидуальной практической работе схемы алгоритма следует написать исходный текст программы на языке Паскаль. В программе предусмотреть вывод на экран всех входных и выходных данных. Программа должна быть хорошо прокомментирована.

Программа должна быть описана. В описании должно быть приведено назначение входных, выходных и внутренних переменных, назначение основных блоков программы.

На основании разработанного исходного текста программы рассчитываются значения метрик Холстеда:

* шесть базовых метрик Холстеда (результаты должны быть сведены в таблицу, аналогичную таблице 2);
* словарь программы;
* длина программы;
* объем программы;

При анализе исходного текста программы следует пользоваться таблицей 1, представленной в теоретических сведениях к лабораторным работам.

На основании разработанного исходного текста программы рассчитываются значения метрик сложности потока данных:

* спены идентификаторов и суммарный спен программы (результаты должны быть сведены в таблицу, аналогичную таблице 8);
* полная метрика Чепина и метрика Чепина ввода/вывода (результаты должны быть сведены в таблицу, аналогичную таблице 9 с соответствующими пояснениями по распределению переменных по группам).

Содержание отчета по индивидуальной практической работе № 2

Индивидуальная практическая работа № 2 должна содержать:

* титульный лист (образец титульного листа приведен выше);
* номер и условие индивидуального задания;
* исходный текст программы на языке Паскаль с комментариями;
* описание программы;
* расчет метрик Холстеда ддя разработанной программы с результатами, представленными в виде таблицы;
* расчет спена разработанной программы с результатами, представленными в виде таблицы;
* расчет полной метрики Чепина и метрики Чепина ввода/вывода с результатами, представленными в виде таблицы.