

Министерство образования Российской Федерации
Международный образовательный консорциум
«Открытое образование»
Московский государственный университет экономики,
статистики и информатики
АНО «Евразийский открытый институт»

Е.В. Ковалевская

**Метрология, качество и
сертификация
программного обеспечения**

Учебная программа
Руководство по изучению дисциплины
Учебное пособие
Практикум по дисциплине
Тестовые задания по дисциплине

Москва 2004

УДК 659,4
ББК 60,56
К 981

Ковалевская Е.В. МЕТРОЛОГИЯ, КАЧЕСТВО И СЕРТИФИКАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ: Учебная программа, руководство по изучению дисциплины, учебное пособие, практикум по курсу, тестовые задания по дисциплине – Московский государственный университет экономики, статистики и информатики, М., 2004, – 96 с.

Учебно-методические материалы предназначены для студентов, обучающихся по специальностям «Прикладная информатика в экономике» и «Математическое обеспечение и администрирование информационных систем», а также может быть рекомендовано студентам других специальностей при изучении вопросов, связанных с метрологией, качеством и сертификацией программного обеспечения.

© Ковалевская Е.В., 2004 г.
© Московский государственный университет экономики,
статистики и информатики, 2004 г.

СОДЕРЖАНИЕ

Учебная программа по дисциплине «Метрология, качество и сертификация программного обеспечения».....	9
Руководство по изучению дисциплины «Метрология, качество и сертификация программного обеспечения».....	15
Учебная пособие	
«Метрология, качество и сертификация программного обеспечения».....	29
1. Метрология.....	30
2. Критерии качества комплексов программ	30
3. Схема взаимодействия основных критериев качества программ.....	33
4. Зависимость эффективности и затрат на единицу времени от этапов жизненного цикла комплексов программ	33
5. Корректность программных средств	35
6. Основные виды корректности комплексов программ	36
7. Схема взаимодействия компонент, определяющих обнаруживаемые отклонения программ эталонов.....	37
8. Типы эталонов и методы проверки корректности программ.....	38
9. Методы получения эталонных значений.....	39
10. Блок-схема системы верификации программных модулей.....	39
11. Общая схема отладки программы.....	40
12. Классификационная схема ошибок	41
13. Сложность программного обеспечения.....	42
14. Основные виды сложности	43
15. Схема взаимодействия показателей вычислительной сложности и основные факторы, влияющие на их значения.....	44
16. Определение характеристик сложности.....	44
17. Надежность программных средств	45
18. Сбор данных, необходимых для расчета матрицы вероятностей Р.....	45
19. Эффективность	47
20. Разработка программных средств	48
21. Факторы, определяющие затраты на создание ПС	49
22. Сертификация	51
Практикум по дисциплине «Метрология, качество и сертификация программного обеспечения».....	59
Тестовые задания по дисциплине «Метрология, качество и сертификация программного обеспечения».....	83

Учебная программа

1. Цели и задачи дисциплины, ее место в учебном процессе

Курсы «Метрология и сертификация программного обеспечения», «Метрология и качество программного обеспечения» читаются студентам в 4 семестре и базируется на предшествующих дисциплинах системного и прикладного программирования.

Данный курс призван формировать у студентов теоретические знания и практические навыки по проблемам оценки качества и повышения надежности программного обеспечения.

Студенты в результате изучения курса должны:

- знать: задачи и методы исследования надежности и качества программных средств (ПС); классификационную схему программных ошибок, средства и методы разработки надежного программного обеспечения;
- уметь: проектировать, конструировать и отлаживать программные средства с заданными критериями качества;
- выявлять основные факторы, определяющие качество и надежность программных средств;
- осуществлять моделирование требуемого уровня надежности в соответствии с заданными критериями;
- оценивать технико-экономические показатели разработки ПС.

Перечень дисциплин, усвоение которых необходимо для изучения курса «Метрология и сертификация программного обеспечения» и «Метрология и качество программного обеспечения».

№ п/п	Наименование дисциплины
1.	Основы алгоритмизации и алгоритмические языки
2.	Разработка и стандартизация программных средств и информационных технологий.
3.	Объектно-ориентированное программирование

2. Содержание дисциплины

В методическом отношении курс включает лекции, практические и лабораторные занятия, а так же самостоятельную работу студентов. После обучения студенты сдают экзамен по курсу.

2.1. Лекционные занятия

В лекциях излагается основное содержание дисциплины на современном этапе его развития.

Содержание лекционного курса.

Введение.

Предмет, содержание и задачи курса, связь с другими дисциплинами.

Роль курса в формировании специалиста в соответствии с его квалификационной характеристикой.

Основные понятия и определения.

Тема 1. Метрология программных средств.

Метрология – наука об измерениях. Основные понятия и термины. История развития метрологии. Закон «Об обеспечении единства измерений». Основы метрологической оценки программных средств.

Тема 2. Качество программных средств.

Задача количественной оценки качества ПС.

Виды метрик: интервальные, порядковые и категориальные шкалы.

Показатели качества ПС: сложность, корректность, надежность, трудоемкость.

Стандарты, регламентирующие показатели качества ПС. Выбор и измерение показателей качества на основных этапах жизненного цикла ПС.

Применение метрик в управлении качеством ПС. Инструментальные, программные и аппаратные средства измерений и количественной оценки качества ПС.

Тема 3. Сложность программных средств.

Основные виды сложности проектирования и функционирования ПС.

Показатели вычислительной сложности: временная, программная, информационная сложность и основные факторы, влияющие на их значение.

Измерение и оценка сложности программных средств.

Тема 4. Корректность программных средств.

Основные понятия и виды корректности программ. Функциональная, детерминированная, стохастическая, динамическая корректность.

Типы эталонов, методы измерений и проверки корректности программ. Ошибки в ПС. Количественное описание ошибок ПС. Классификационная схема программных ошибок.

Источники ошибок. Применение метрики ПС для обнаружения и устранения ошибок.

Тема 5. Надежность программных средств.

Определение надежности ПС. Показатели надежности ПС. Факторы, определяющие надежность ПС. Определение показателей надежности на различных этапах жизненного цикла ПС. Аналитические, имитационные, экспериментальные методы оценки надежности ПС. Моделирование и обеспечение надежности в процессе создания ПС. Статистические, динамические, эмпирические модели.

Тема 6. Технико-экономические показатели разработки программных средств.

Цели технико-экономического анализа разработки ПС. Составляющие затрат на разработку ПС. Факторы, определяющие затраты на создание ПС. Методы сбора и обработки данных о разработках ПС. Трудоемкость, длительность, стоимость разработки ПС. Экономическая эффективность ПС.

Тема 7. Сертификация программных средств.

Задачи и проблемы сертификации ПС. Виды сертификационных испытаний программ. Методы, технология, средства обеспечения сертификации ПС. Стандарты сертификации ПС.

Заключение по курсу.

2.2. Практические занятия

Практические занятия проводятся в учебных группах с целью закрепления пройденного теоретического материала, излагаемого в лекциях. Практические занятия по каждой теме проводятся в соответствии с планом распределения времени.

Материал, освоенный на практических занятиях, позволяет выполнить лабораторную работу.

Контроль знаний студентов на практических занятиях проверяется путем проведения аудиторных контрольных работ, а также опроса студентов, показывающего степень усвоения ими материала.

Самостоятельная работа студентов заключается в изучении методов и средств, обеспечивающих оценку качества и надежности ПС, моделирования требуемого уровня надежности ПС с заданными критериями.

2.3. Лабораторная работа

№ п/п	Содержание работы	Объем в часах	
		аудиторн. занятия	самост. работа
1.	Оценка качественных и технико-экономических показателей программных средств	2	8

2.4. Распределение бюджета времени (в часах)

№ темы	Всего	в том числе		
		лекции	практ. занят.	лаб. раб.
Введение	2	2	–	–
Тема 1. Метрология ПС	2	2		
Тема 2. Качество ПС	14	6	8	–
Тема 3. Сложность ПС	6	4	2	–
Тема 4. Корректность ПС	10	6	4	–
Тема 5. Надежность ПС	20	6	10	4
Тема 6. Техничко-экономические показатели разработки ПС	8	4	4	–
Тема 7. Сертификация ПС	2	2	–	–
	64	32	28	4

2.5. Литература

Основная

- 1.Благодатских и др. «Экономика, разработка и использование ПО ЭВМ». М., «Финансы и статистика», 1995г.
- 2.Сертификация продукции. Международные стандарты и руководства ИСО/МЭК в области сертификации и управления качеством. М., издательство стандартов, 1990.
- 3.Ковалевская Е.В., Енгибарян М.А. «Экономическая эффективность и надежность программных средств вычислительной техники» М., МЭСИ, 1990.

Дополнительная

- 1.Боэм Б.У. «Инженерное проектирование программного обеспечения» перевод с англ. / под редакцией А.А. Красилова, М. Радио и связь; 1985.
- 2.Шураков В.В. «Надежность программного обеспечения систем обработки данных», М. Финансы и статистика, 1981.
- 3.Е.Я. Карповский и др. «Надежность программной продукции», Киев. Техника, 1990.
- 4.Липаев В.В. «Надежность программных средств», Синтег, М..1998.
- 5.Костогрызов А.И., Липаев В.В. «Сертификация качества функционирования автоматизированных информационных систем»,М., издательство Вооружение.Политика. Конверсия, 1995.
- 6.Щербо В.К., Козлов В.А. «Функциональные стандарты в открытых системах» 2.1. М., издательство МЦНТИ, 1997.

Руководство по изучению дисциплины

1. Цель курса

Приобретение студентами знаний, формирование умений и навыков в области метрологии, качества и сертификации программных средств (ПС).

Задачи:

Студенты в результате изучения курса должны:

– **Освоить:** систему понятий и терминов метрологии и сертификации ПС ; правовые основы метрологической деятельности; основы теории измерений ; организационно-методические принципы сертификации ПС на международном, региональном и национальном уровнях.

– **Знать:** задачи и методы исследования качества и надежности ПС; классификационную схему программных ошибок, средства и методы разработки надежного программного обеспечения.

– **Уметь:** проектировать, конструировать и отлаживать программные средства с заданными критериями качества;

выявлять основные факторы, определяющие качество и надежность программных средств;

осуществлять моделирование требуемого уровня надежности в соответствии с заданными критериями;

оценивать технико-экономические показатели разработки ПС.

Сфера профессионального использования

Маркетинговые и коммерческие исследования качества программной продукции с использованием знаний в области метрологии и сертификации. Разработка, оценка, контроль и аттестация программных средств.

2. Для изучения данной дисциплины студент должен знать:

- основы алгоритмизации и алгоритмические языки;
- разработку и стандартизацию программных средств и информационных технологий;
- объектно-ориентированное программирование;
- операционные среды, системы и оболочки;
- численные методы;
- методы оптимизации.

В методическом отношении курс включает лекции, практические и лабораторные занятия, выполнение курсовой работы, а так же самостоятельную работу студентов. После обучения студенты сдают экзамен по курсу.

Лекционные занятия

В лекциях излагается основное содержание дисциплины на современном этапе его развития.

Практические занятия

Практические занятия проводятся в учебных группах с целью закрепления пройденного теоретического материала, излагаемого в лекциях. Практические занятия по каждой теме проводятся в соответствии с планом распределения времени.

Материал, освоенный на практических занятиях, позволяет выполнить лабораторную работу.

Контроль знаний студентов на практических занятиях проверяется путем проведения аудиторных контрольных работ, а также опроса студентов, показывающего степень усвоения ими материала.

Самостоятельная работа студентов заключается в изучении методов и средств, обеспечивающих оценку качества и надежности ПС, моделирования требуемого уровня надежности ПС с заданными критериями.

Лабораторная работа

№ п/п	Содержание работы	Объем в часах	
		аудиторн. занятия	самост. работа
1.	Оценка качественных и технико-экономических показателей программных средств	2	8

3. Перечень основных тем

Введение.

Предмет, содержание и задачи курса, связь с другими дисциплинами.

Роль курса в формировании специалиста в соответствии с его квалификационной характеристикой.

Основные понятия и определения.

Во введении должны быть освещены вопросы:

– Метрология – наука о получении измерительной информации, ее задачи и роль в народном хозяйстве страны;

– Д.И. Менделеев – основоположник русской метрологии;

– Роль метрологии и сертификации программных средств в обеспечении их качества;

– Назначение и содержание дисциплины;

– Взаимосвязь стандартизации метрологии и сертификации программных средств с другими областями знаний и производства;

– Роль и место курса в процессе подготовки специалистов.

Основные понятия: метрология, стандартизация, сертификация, программное средство, измерения, ранжирование, качество, надежность, эффективность.

Тема 1. Метрология ПС

Цель изучения – получение знаний по проблемам метрологического обеспечения программных средств.

Основы метрологического обеспечения программных средств.

Роль метрологического обеспечения в повышении качества ПС.

История развития метрологии.

Метрология в зарубежных странах и международные метрологические организации.

Государственная метрологическая служба в РФ.

Закон « Об обеспечении единства измерений».

Изучив данную тему, студент должен ответить на следующие вопросы:

1. В чем заключаются основные задачи метрологии?
2. На какие виды подразделяется метрология?
3. Что понимают под измерением?
4. Какие измерения называют прямыми, косвенными, совокупными, совместными?
5. Какие Вы знаете средства измерений?
6. Что такое поверка средств измерений?
7. Дайте характеристику существующих методов измерений?
8. Что понимают под метрологической службой?
9. Кто был одним из основоположников метрологии в России?
10. Какова роль метрологии в повышении качества ПС?

Тема 2. Качество программных средств

Цель изучения – получение знаний по проблемам качества программных средств и их оценки.

Задача количественной оценки качества ПС.

Виды метрик: интервальные, порядковые и категориальные шкалы.

Показатели качества ПС: сложность, корректность, надежность, трудоемкость.

Стандарты, регламентирующие показатели качества ПС. Выбор и измерение показателей качества на основных этапах жизненного цикла ПС.

Применение метрик в управлении качеством ПС. Инструментальные, программные и аппаратные средства измерений и количественной оценки качества ПС.

Изучив данную тему, студент должен ответить на следующие вопросы:

1. Структура и жизненный цикл программных средств.
2. Измерение качества программных средств, виды метрик.
3. Основные критерии качества программных средств.
4. Факторы, определяющие качество программных средств.
5. Выбор и измерение показателей качества программных средств.
6. Международные и отечественные стандарты, регламентирующие качество программных средств.
7. Показатели качества без данных.
8. Обеспечение качества программных средств в процессе разработки программных средств.
9. Управление качеством программных средств.
10. Средства измерений и оценки программных средств.

План практического занятия по теме 2

1. Методика оценки качественных показателей программных средств.
2. Подготовка к выполнению лабораторной работы (1-ая часть – оценка качественных показателей программных средств.)

Тема 3. Сложность программных средств

Цель изучения – получение необходимых знаний о сложности программных средств и методах её оценки.

Основные виды сложности проектирования и функционирования программных средств.

Показатели вычислительной сложности: временная, программная, информационная сложность и основные факторы, влияющие на их значение.

Измерение и оценка сложности программных средств.

Изучив данную тему, студент должен ответить на следующие вопросы:

1. Понятие сложности программы, комплекса программ.
2. Основные виды сложности проектирования и функционирования программных средств.
3. Временная, программная, информационная сложность программных средств и их оценка.
4. Оценка структурной сложности программных средств.
5. Определение статистической сложности программных средств.

Тема 4. Корректность программных средств

Цель изучения – приобретение более глубоких знаний по методам оценки корректности программных средств.

Основные понятия и виды корректности программ. Функциональная, детерминированная, стохастическая, динамическая корректность.

Типы эталонов, методы измерений и проверки корректности программ.

Ошибки в программных средствах. Количественное описание ошибок.

Классификационная схема программных ошибок.

Источники ошибок. Применение метрики ПС для обнаружения и устранения ошибок.

Изучив данную тему, студент должен ответить на следующие вопросы:

1. Основные виды корректности программных средств (корректность текстов программ, программных модулей, данных, комплексов программ).
2. Программные эталоны.
3. Методы проверки корректности программ.
4. Верификация программ.
5. Характеристики программных ошибок.
6. Понятие ошибки в программе.
7. Разновидности математических моделей распределения ошибок в программных комплексах.
8. Классификационная схема программных ошибок.
9. Источники ошибок.
10. Принципы, стратегии и этапы тестирования и отладки программных средств.
11. Средства автоматизации тестирования программных средств.
12. Регистрация и обработка результатов испытаний программных средств.

План практического занятия по теме 4.

1. Тестирование и отладка программных средств. Разработка тестовых наборов данных.
2. Ошибки в ПС. Анализ корректности тестирования ПС.
3. Подготовка к выполнению лабораторной работы (2-ая часть – разработка тестовых наборов данных, тестирование, сбор и классификация данных об ошибках.)

Тема 5. Надёжность программных средств

Цель изучения – получение необходимых теоретических и практических знаний по методам оценки надёжности программных средств.

Определение надёжности программных средств.

Показатели надёжности программных средств.

Факторы, определяющие надёжность программных средств.

Определение показателей надёжности на различных этапах жизненного цикла программных средств.

Аналитические, имитационные, экспериментальные методы оценки надёжности программных средств.

Моделирование и обеспечение надёжности в процессе создания программных средств.

Статические, динамические, эмпирические модели надёжности.

Изучив данную тему, студент должен ответить на следующие вопросы:

1. Основные понятия и показатели надёжности программных средств.
2. Факторы и методы обеспечения надёжности разработки и функционирования программных средств.
3. Особенности обеспечения надёжности функционирования импортных программных средств.
4. Моделирование и обеспечение надёжности в процессе создания программных средств.
5. Оценка надёжности программных средств с использованием статических и динамических моделей.
6. Методы обеспечения технологической безопасности программных средств и данных.

План практического занятия по теме 5

1. Решение задач по оценке надёжности программных средств (статические и динамические модели надёжности).
2. Подготовка к выполнению лабораторной работы (3-ья часть – оценка надёжности программных средств).

Тема 6. Техничко-экономические показатели разработки программных средств

Цель изучения – получение знаний по экономическим аспектам разработки и эксплуатации программных средств.

Цели технико-экономического анализа разработки программных средств.

Составляющие затрат на разработку программных средств.

Факторы, определяющие затраты на создание программных средств.

Методы сбора и обработки данных о разработках программных средств.

Трудоёмкость, длительность и стоимость разработки программных средств. Экономическая эффективность программных средств.

Изучив данную тему, студент должен ответить на следующие вопросы:

1. Основные задачи исследования затрат на создание программных средств.
2. Техничко-экономические характеристики разработки программных средств.
3. Факторы, определяющие затраты на создание программных средств.

4. Оценка трудоёмкости, стоимости, длительности разработки программных средств.
5. Методы сбора и обработки данных о разработках программных средств.
6. Расчет показателей экономической эффективности программных средств.

План практического занятия по теме 6

1. Расчет технико-экономических показателей разработки программных средств.
2. Подготовка к выполнению лабораторной работы (4-ая часть – оценка технико-экономических показателей разработки программных средств).

Тема 7. Сертификация программных средств

Цель изучения – приобретение знаний по сертификации программных средств.

Задачи и проблемы сертификации программных средств.

Виды сертификационных испытаний программ.

Методы, технология, средства обеспечения сертификации программных средств.

Стандарты сертификации.

Изучив данную тему, студент должен ответить на следующие вопросы:

1. Сущность, значение и задачи сертификации программных средств.
2. Роль сертификации программных средств в повышении её качества и конкурентоспособности.
3. Основные понятия и определения в области сертификации: сертификация, сертификат соответствия, испытательная лаборатория.
4. Виды сертификационных испытаний программных средств и баз данных (добровольная, обязательная сертификация).
5. Организация сертификационных испытаний программных средств и баз данных.
6. Содержание протокола испытаний программных средств.
7. Аккредитация испытательных лабораторий, достоверность сертификационных испытаний.

План практического занятия по теме 7

Знакомство с основными нормативными документами Российской системы сертификации, со стандартами, регламентирующими организацию сертификации качества информационных систем, программных средств и баз данных.

4. Для проведения итогового контроля необходимо подготовиться к экзамену по следующим теоретическим вопросам

1. Специфические особенности ПС ВТ. ПС – новый вид товарной продукции.
2. Жизненный цикл ПС. Содержание основных этапов жизненного цикла ПС.
3. Анализ и разработка требований к ПС.
4. Определение целей создания ПС.
5. Разработка внешних спецификаций на ПС.
6. Цели и порядок внутреннего проектирования ПС.
7. Модульная структура ПС.
8. Внешнее проектирование модулей.
9. Проектирование и кодирование модулей.
10. Стиль программирования.

11. Принципы и методы тестирования ПС.
12. Проектирование теста.
13. Общая характеристика методов тестирования.
14. Тестирование модулей.
15. Тестирование комплексов программ.
16. Отладка программ.
17. Документирование ПС.
18. Испытания и сопровождение ПС.
19. Определение надежности ПС.
20. Показатели надежности ПС.
21. Факторы, определяющие надежность ПС.
22. Общая характеристика моделей надежности ПС.
23. Ошибки ПС.
24. Причины ошибок в ПС.
25. Классификация программных ошибок.
26. Модель Коркорэна.
27. Модель Миллса.
28. Модель простая интуитивная.
29. Модель Муса.
30. Модель Шумана.
31. Модель Нельсона.
32. Модель переходных вероятностей.
33. Модель Гоэл-Окимото.
34. Модель Джелински-Моранды.
35. Модель сложности.
36. Эмпирические модели надежности.
37. Динамические модели надежности.
38. Статические модели надежности.
39. Особенности определения экономической эффективности ПС.
40. Расчет экономического эффекта при производстве ПС.
41. Расчет экономического эффекта при применении ПС.
42. Расчет коэффициента экономической эффективности капитальных вложений и срока окупаемости капитальных вложений ПС.
43. Цели технико-экономического анализа разработки ПС.
44. Факторы, определяющие затраты на создание ПС.
45. Составляющие затрат на разработку ПС.
46. Методы сбора и обработки данных о разработках ПС.
47. Трудоемкость, длительность, стоимость разработки ПС.
48. Методы обеспечения технологической безопасности ПС и данных.
49. Задачи и проблемы сертификации ПС.
50. Виды сертификационных испытаний программ.
51. Методы, технология, средства обеспечения сертификации ПС.
52. Стандарты сертификации ПС.
53. Задача количественной оценки качества ПС. Виды метрик.
54. Показатели качества ПС. Стандарты, регламентирующие показатели качества ПС.
55. Выбор и измерение показателей качества ПС.
56. Понятие сложности, основные компоненты сложности.
57. Показатели вычислительной сложности.

58. Оценка сложности ПС.
59. Основные понятия и виды корректности программ.
60. Типы эталонов, методы измерений и проверки корректности программ.

**Тематика курсовых работ по дисциплине
«Метрология, качество и сертификация программного обеспечения»**

1. Основы метрологии программных средств. Роль метрологии в повышении качества программных средств
2. История русской метрологии XI-XXв.
3. Метрология и научно-технический прогресс.
4. Применение метрик в управлении качеством программных средств.
5. Характеристика показателей качества программных средств.
6. Методы определения численных значений показателей качества программных средств.
7. Сложность проектирования и функционирования программных средств.
8. Оценка структурной сложности программных средств.
9. Корректность программных средств и методы ее измерения.
10. Ошибки в программных средствах.
11. Применение метрики программных средств для обнаружения и устранения ошибок.
12. Верификация и синтез программ.
13. Методы испытаний программ на надежность.
14. Техничко-экономические показатели разработки программных средств и их оценка.
15. Эмпирические модели надежности программных средств.
16. Динамические модели надежности программных средств.
17. Статические модели надежности программных средств.
18. Моделирование и обеспечение надежности в процессе создания программных средств.
19. Роль сертификации программных средств в повышении их качества и конкурентоспособности.
20. Нормативное обеспечение сертификации программных средств.
21. Сертификация программных средств, услуг, защита прав потребителей.
22. Стандартизация программных средств и ее место в управлении качеством программных средств.
23. Органы и службы стандартизации в РФ.
24. Международная стандартизация, ее роль в мировой экономике.
25. Разработка и оценка программных средств задач предметной области (предметная область выбирается студентом в соответствии с его специализацией).

5. Список литературы и ссылки на ресурсы Интернет

1. Метрология

- 1) <http://www.oilspace.ru/metrolog/gost/osnova.htm>
- 2) <http://www.standard.ru/>
- 3) <http://www.iso9000.boom.ru/docs/docs.html>
- 4) http://k46.aanet.ru/textbooks/std_pro/index.htm
- 5) http://k46.aanet.ru/textbooks/std_pro/index1_2.htm
- 6) http://k46.aanet.ru/textbooks/std_pro/index2_1.htm
- 7) http://k46.aanet.ru/textbooks/std_pro/index2_2.htm

2. Качество

- 1) Липаев В.В. Качество программного обеспечения. – М.: Финансы и статистика, 1983 (стр. 7-30. Основные показатели качества программ.)
- 2) Липаев В.В. Выбор и оценивание характеристик качества программных средств. М., Синтег, 2001.
- 3) http://k46.aanet.ru/textbooks/std_pro/pril_st.htm
- 4) http://k46.aanet.ru/textbooks/std_pro/index1_4.htm
- 5) http://k46.aanet.ru/textbooks/std_pro/index1_3.htm
- 6) <http://www.udc.com.ua/dc/udc?p=294#1>
- 7) <http://www.udc.com.ua/dc/udc?p=554>
- 8) <http://www.udc.com.ua/dc/udc?p=288>
- 9) <http://www.udc.com.ua/dc/udc?p=284>
- 10) <http://www.udc.com.ua/dc/udc?p=174>
- 11) <http://www.swquality.com>

3. Надежность

- 1) Благодатских и др. «Экономика, разработка и использование ПО ЭВМ» М., «Финансы и статистика», 1995г.
- 2) Ковалевская Е.В., Енгибарян М.А. «Экономическая эффективность и надежность программных средств вычислительной техники» М., МЭСИ, 1990
- 3) Шураков В.В. «Надежность программного обеспечения систем обработки данных», М. Финансы и статистика, 1981.
- 4) <http://inf.susu.ac.ru/~tyrty/nadejnost.html> А.В. Коганов, С.Г. Романюк Экономический подход к надежности программ.
- 5) <http://b-test.narod.ru/index.htm>.
- 6) <http://b-test.narod.ru/what.htm>.
- 7) Липаев В.В. Надежность программных средств. Издательство «Синтег» 2001.
- 8) Липаев В.В. Качество программного обеспечения. 1983. Финансы и Статистика (стр. 215-258. Надежность комплексов программ)
- 9) http://www.creativeport.ru/programming/tv_doc/tvdoc_06.html
- 10) <http://www.lib.com.ua/comp/26.34.html>
- 11) <http://lib.promedia.minsk.by/data/shpargalka/belarus/programming/pro-030.htm>
- 12) Гласс Р. Руководство по надежному программированию: Пер. с англ.-М.: Финансы и статистика, 1982.
- 13) Шураков В.В. Надежность программного обеспечения систем обработки и данных: Учеб.-2-е изд., перераб. и доп.-М.: Финансы и статистика, 1987

4. Корректность

- 1) Липаев В.В. Качество программного обеспечения. 1983. Финансы и Статистика. (стр. 126-214, Корректность комплексов программ).
- 2) <http://www.aha.ru/~nike/Rus/Develop/Ch5.htm>
<http://lib.promedia.minsk.by/data/shpargalka/belarus/programming/pro-030.htm>
- 3) <http://se.math.spbu.ru/Courses/Testing/ModuleTesting.html>
- 4) <http://www.otd.tstu.ru/~education/direct2/kurs/Cpp/G3/36.html>

- 5) Алагич С., Арбиб М. Проектирование корректных структурированных программ: Пер. с англ.-М.: Радио и связь, 1984
- 6) Бейбер Р.Л. Программное обеспечение без ошибок: приемы и секреты создания правильных программ. Пер. с англ.-М.: Джон Уайли энд Санз, 1996.

5. Сложность

- 1) Деметрович Я. и др. Автоматизированные методы спецификации: Пер. с англ./Деметрович Я., Кнут Е., Радо П.-М.: Мир, 1989.
- 2) <http://economy.mari.ru/test/oalp/LECTIION/204.htm>
- 3) http://k46.aanet.ru/textbooks/std_pro/index1_4.htm
- 4) Липаев В.В. Качество программного обеспечения. 1983. Финансы и Статистика. (стр. 67-125, Сложность комплексов программ).
- 5) <http://www.junik.lv/~georg/programing/sa/ocenska.htm>

6. Эффективность

- 1) Благодатских и др. «Экономика, разработка и использование ПО ЭВМ» М., «Финансы и статистика», 1995г.
- 2) Ковалевская Е.В., Енгибарян М.А. «Экономическая эффективность и надежность программных средств вычислительной техники» М., МЭСИ, 1990
- 3) <http://www.otd.tstu.ru/~education/direct2/kurs/Cpp/G3/36.html>
- 4) Ван Тассел Д. Стиль, разработка, эффективность, отладка и испытание программ: Пер.с англ.-2-е изд., испр.-М.: Мир, 1985.
- 5) <http://parallel.rb.ru/tech/opt/>

7. Сертификация

- 1) Сертификация продукции. Международные стандарты и руководства ИСО/МЭК в области сертификации и управления качеством. М., Издательство стандартов, 1990.
- 2) <http://www.standard.ru/iso9000/iso9000.phtml>
- 3) <http://www.iso9000.boom.ru/docs/docs.html>
- 4) http://www.osp.ru/os/2000/10/048_print.htm
- 5) Угрюмова В.Н., Позднеев Б.М., Васильев В.В. Проблемы и перспективы сертификации программного обеспечения. Сертификация, 1997 г., № 2,
- 6) Липаев В.В. Сертификация информационных технологий и баз данных. Методы и стандарты. Казань, 1995

6. Глоссарий

Аккредитация	Единицы измерения:
Адаптируемость	– объёма программ
Безопасность	– технико-экономических показателей
Верификация программ	Допуск ошибок
Вероятность сбоя	Жизненный цикл программных средств
Виды метрик	Заданный уровень надёжности
Внешние спецификации	Затраты:
Длина программы	– на создание ПС

- при эксплуатации ПС
- на сопровождение ПС

Защита от ошибок
Исправление ошибок
Испытание
Источники ошибок
Информативность
Качество программных средств
Критерии качества программных средств
Корректность программных средств:

- структурная
- функциональная
- детерминированная
- стохастическая
- динамическая

Коэффициент изменения трудозатрат
Критерии эффективности программных средств
Концептуальная целостность
Метрология программных средств
Модель:

- оценки надёжности ПС
- сложности ПС

Модульная структура
Надёжность программных средств

Ошибка
Обнаружение ошибок
Сложность программных средств:

- проектирования
- функционирования
- программных модулей
- структурная
- статистическая
- вычислительная
- программная
- информационная

Технико-экономические показатели:

- длительность
- производительность труда
- трудоёмкость

Экономика программных средств
Эффективность
Экономический эффект
Коэффициент экономической эффективности капитальных вложений
Срок окупаемости
Сертификация
Сертификат соответствия
Лаборатория сертификации

Учебное пособие

1. Метрология



Метрология – это наука об измерениях, методах, средствах обеспечения их единства и способах достижения требуемой точности.

Предметом метрологии является извлечение количественной информации о свойствах объектов и процессов с заданной точностью и достоверностью.

Средства метрологии – это совокупность средств измерений и метрологических стандартов, обеспечивающих их рациональное использование.

Измерение – это нахождение значения физической величины опытным путем с помощью специальных технических средств.

Ранжирование – это распределение величин по возрастающим или убывающим показателям, характеризующим те или иные свойства этой величины.

Метрологическое обеспечение – это установление и применение научных и организационных основ, технических средств, правил и норм, необходимых для достижения единства и требуемой точности измерений.

2. Критерии качества комплексов программ



Критерии качества представляют собой измеряемые численные показатели в виде некоторой целевой функции, характеризующие степень выполнения программами своего назначения.

Специалисты стремятся каждый раз выделить некоторый превалирующий показатель для оценки качества системы, к которому предъявляются следующие основные требования:

- критерий должен численно характеризовать степень выполнения основной целевой функции системы, наиболее важной для данного этапа анализа или синтеза;
- критерий должен обеспечивать возможность определения затрат, необходимых для достижения его различных значений, а также степени влияния на показатель качества различных внешних факторов и параметров;
- критерий должен быть по возможности простым по содержанию, хорошо измеряемым и иметь малую дисперсию, т. е. слабо зависеть от множества неконтролируемых факторов.

Применение метрик – числовых оценок параметров к комплексам программ позволяет упорядочить их разработку, испытания, эксплуатацию и сопровождение.

Функциональные критерии отражают основную специфику применения и степень соответствия программ их целевому назначению.

Конструктивные критерии качества программ достаточно инвариантны к их целевому назначению и основным функциям. К ним относятся сложность программ, надежность функционирования, используемые ресурсы ЭВМ, корректность и т.д. В свою очередь конструктивные характеристики комплексов программ целесообразно разделить на **основные критерии (показатели)** качества и факторы или параметры, влияющие на их значения.

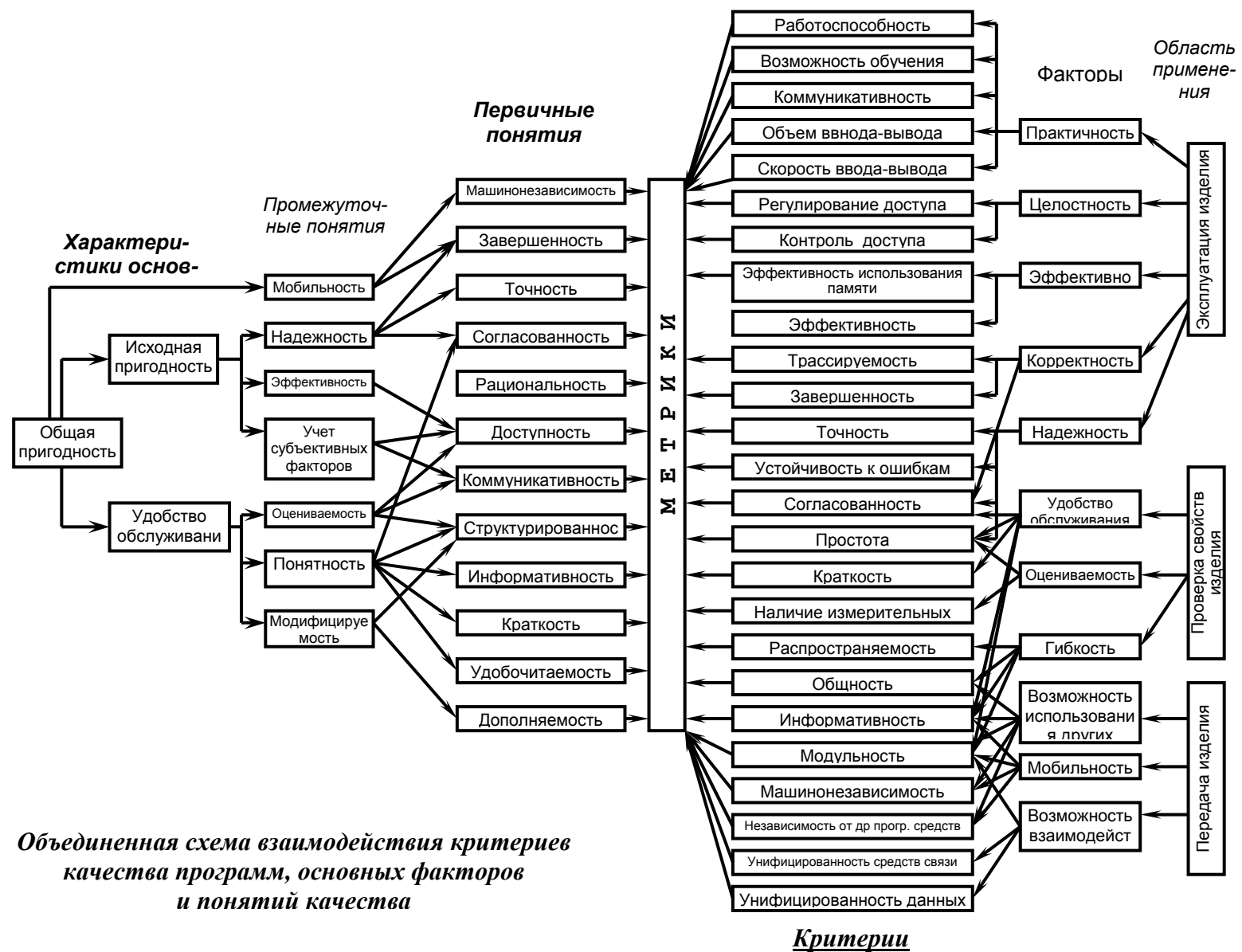
Критерии качества этапа проектирования включают, прежде всего, сложность создания комплекса программ и проверки его адекватности поставленным целям. На этапе проектирования основные затраты составляет трудоемкость создания программ заданной сложности и корректности.

Надежность (безотказность) функционирования характеризует относительную длительность получения корректных (достоверных) результатов или вероятность правильных (не искаженных за допустимые пределы) выходных данных.

Способность к модернизации комплексов программ определяется четкостью их структурного построения и структурой межмодульных связей. Кроме того, на этот критерий влияет метод распределения ресурсов ВС и наличие резервов для развития программ.

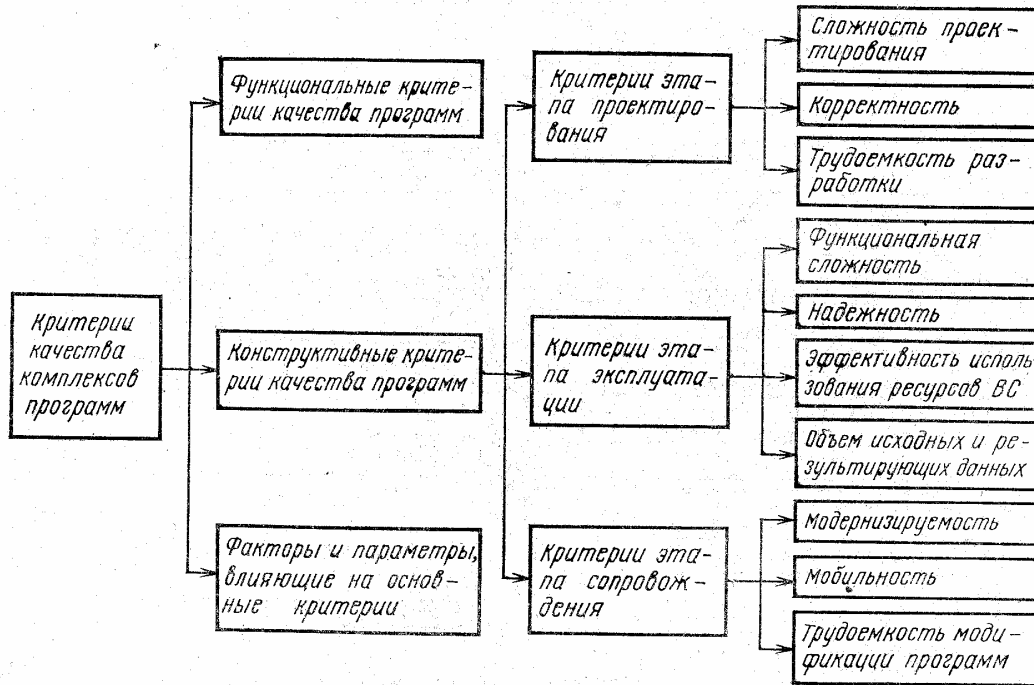
Мобильность комплексов программ относительно изменения типа, структуры и системы команд вычислительной машины характеризует возможность сохранения и эффективного использования эксплуатируемых программ в процессе развития аппаратуры ЭВМ.

Временные показатели жизненного цикла программ: длительность проектирования, продолжительность эксплуатации очередной версии и длительность проведения каждой модификации.

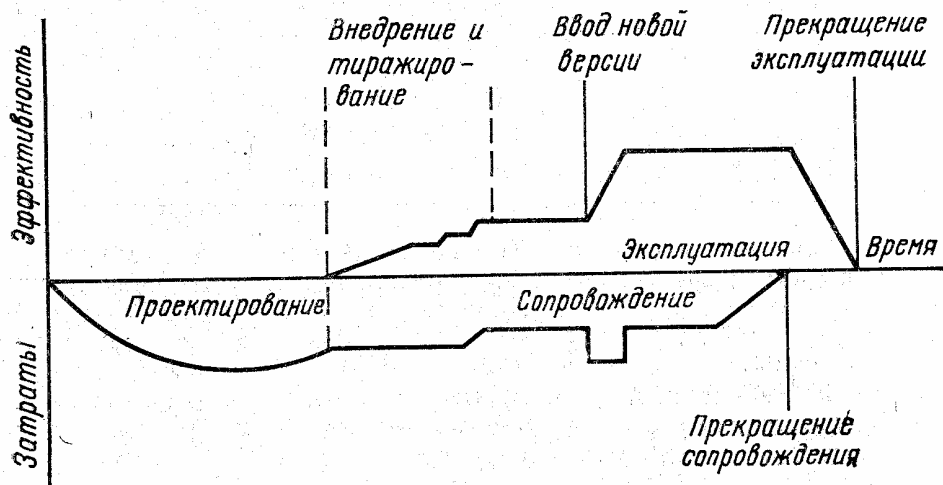


Объединенная схема взаимодействия критериев качества программ, основных факторов и понятий качества

3. Схема взаимодействия основных критериев качества программ



4. Зависимость эффективности и затрат на единицу времени от этапов жизненного цикла комплексов программ





Этапы жизненного цикла	Проектирование	Эксплуатация	Сопровождение
Основные критерии качества комплекса программ	<ol style="list-style-type: none"> 1. Сложность создания программ 2. Корректность программ 3. Трудоемкость разработки программ 	<ol style="list-style-type: none"> 1. Функциональная сложность комплекса программ 2. Надежность функционирования 3. Эффективность использования ресурсов 4. Объем исходных и результирующих данных 	<ol style="list-style-type: none"> 1. Способность к модернизации программ 2. Мобильность программ относительно типов вычислительных систем 3. Трудоемкость изучения и модификации комплексов программ
Основные факторы, определяющие качество	<ol style="list-style-type: none"> 1. Структурная упорядоченность программ и данных 2. Степень стандартизации структуры модулей и переменных 3. Документированность компонент и комплекса 4. Методологическая обеспеченность технологии проектирования 5. Степень комплексной автоматизации технологии проектирования 6. Уровень языков спецификаций, программирования и отладки 7. Квалификация специалистов и методы организации работ 	<ol style="list-style-type: none"> 1. Корректность постановки задач 2. Полнота и точность спецификаций 3. Уровень языков программирования 4. Полнота тестирования программ 5. Степень помехозащищенности программ 6. Документированность для эксплуатации 	<ol style="list-style-type: none"> 1. Структурная упорядоченность комплекса программных средств 2. Степень стандартизации структуры модулей и переменных 3. Документированность для модификации 4. Уровень языков программирования 5. Степень комплексной автоматизации технологии проектирования 6. Обеспеченность контроля изменений версий и распространения копий.

Для анализа требований к качеству, устанавливаемых на этапе выработки требований к системе, существует один полезный метод, основанный на составлении матрицы «требования-свойства». В этой матрице в столбцах располагаются отдельные функциональные требования, а в строках – основные желаемые характеристики качества или свойства программного обеспечения. Возможно и обратное расположение строк и столбцов. Элементы матрицы представляют собой дополнительные функциональные требования, возникающие в ходе детального анализа аспектов качества, связанных с обеспечением каждого необходимого свойства.

<i>Свойства</i>	<i>Требования</i>	
Оцениваемость
Модифицируемость
...



5. Корректность программных средств

Корректность программного средства – соответствие проверяемого объекта некоторому эталонному объекту или совокупности более или менее формализованных эталонных характеристик и правил.

Корректность текстов программ – степень соответствия исходных программ формализованным правилам языков спецификаций и программирования.

Конструктивная корректность модулей – соответствие их структуры общим правилам структурного программирования и конкретным правилам оформления и внутреннего построения программных модулей в данном заказе.

Функциональная корректность модулей – корректность обработки исходных данных и получения результатов.

Конструктивная корректность данных определяется правилами их структурирования и упорядочения.

Функциональная корректность данных связана, в основном, с конкретизацией их содержания в процессе исполнения программ, а также при подготовке данных внешними абонентами.

Конструктивная корректность программных модулей определяется правилами структурного, модульного построения программных комплексов и общими правилами ор-

ганизации межмодульных связей. Эта составляющая может быть проверена формализованными автоматизированными методами.

Функциональная корректность комплексов программ наиболее трудно формируется вследствие большого количества возможных эталонных значений и распределений. В наиболее сложном случае для программ реального времени ее можно разделить на:

- **детерминированную корректность** – должно быть обеспечено однозначное соответствие исходных и результирующих данных исполняемых программ определенным эталонным значениям;
- **стохастическую корректность** – статистическое соответствие распределений результирующих случайных величин заданиям эталонным распределениям при соответствующих распределениях исходных данных;
- **динамическую корректность** – соответствие изменяющихся во времени результатов исполнения программ эталонным данным.

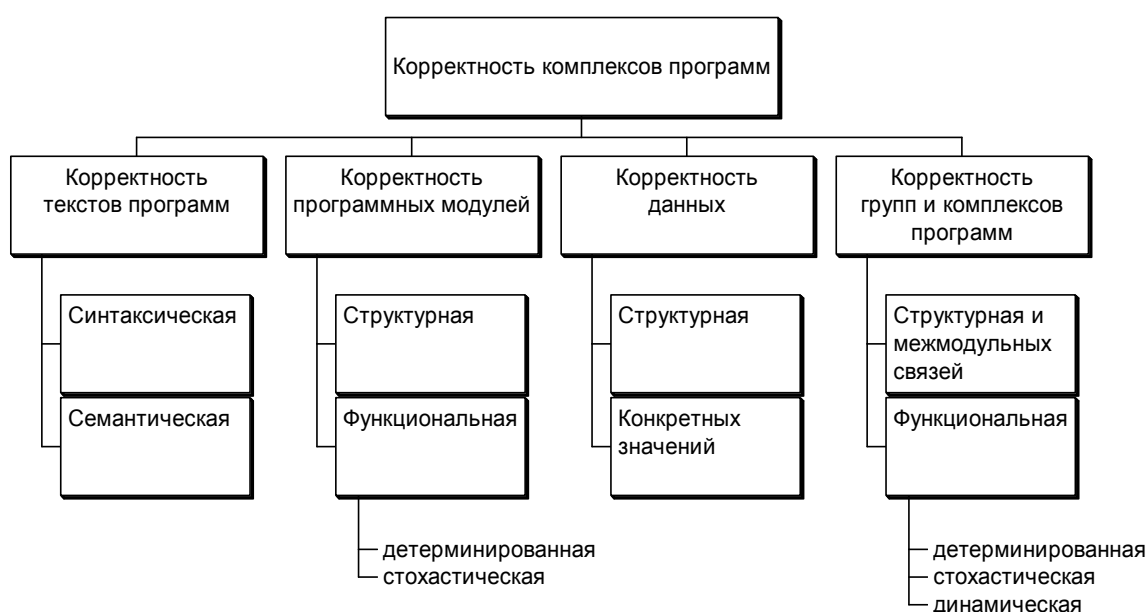
Синтаксический контроль корректности текстов программ – проверка входного текста программ на соответствие синтаксису языка программирования.

Семантический контроль текстов программ – проверка корректности применения и взаимодействия базовых конструкций языка программирования в тексте проверяемых программ.

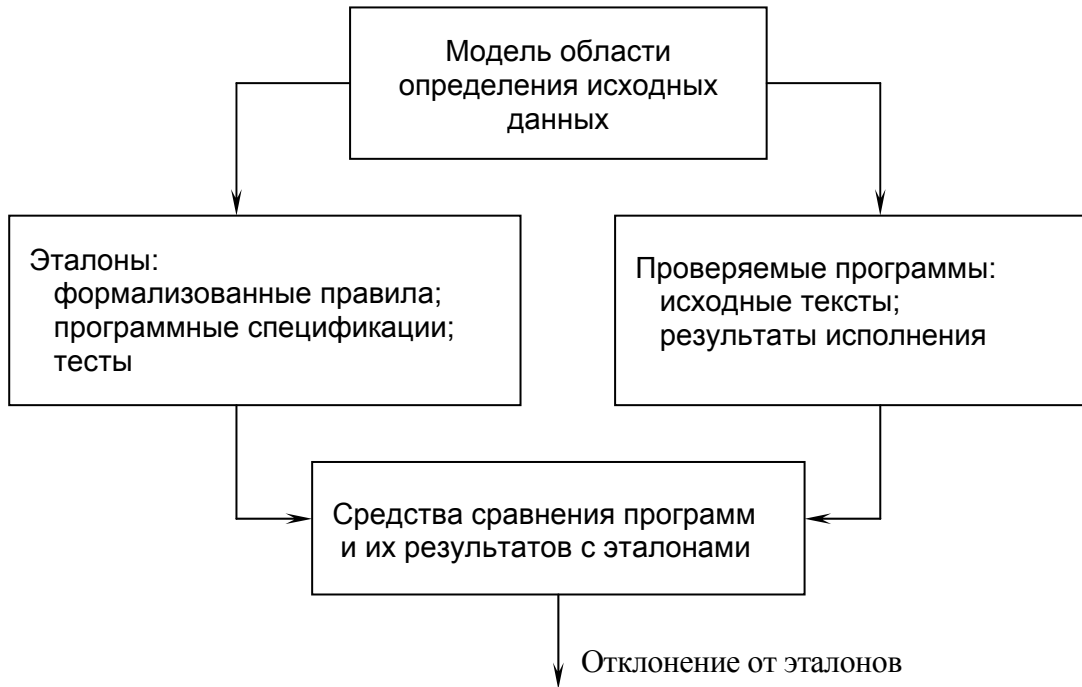
Формализованный структурный контроль программ основывается на статической проверке соответствия структуры программ и последовательности основных операций использования памяти системе эталонных правил.

Верификация (подтверждение правильности) состоит в проверке и доказательстве корректности разработанной программы по отношению к совокупности формальных утверждений, представленных в программной спецификации и полностью определяющих связи между входными и выходными данными этой программы.

6. Основные виды корректности комплексов программ



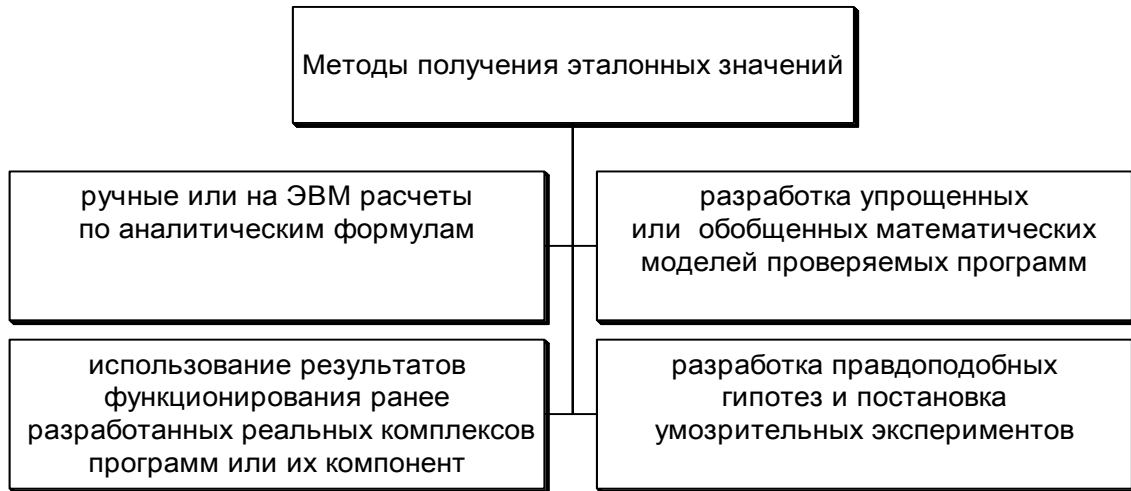
7. Схема взаимодействия компонент, определяющих обнаруживаемые отклонения программ от эталонов



8. Типы эталонов и методы проверки корректности программ



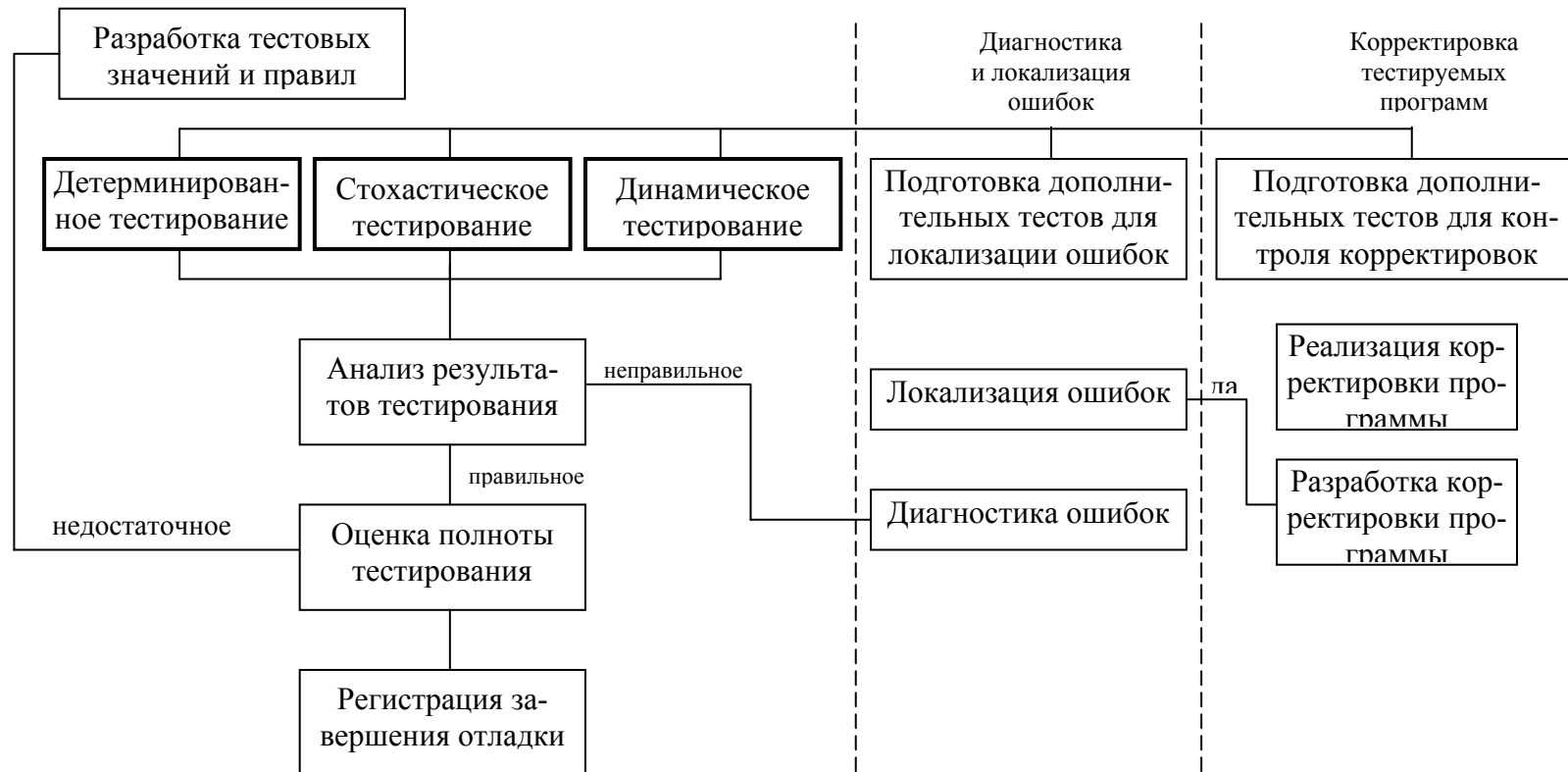
9. Методы получения эталонных значений



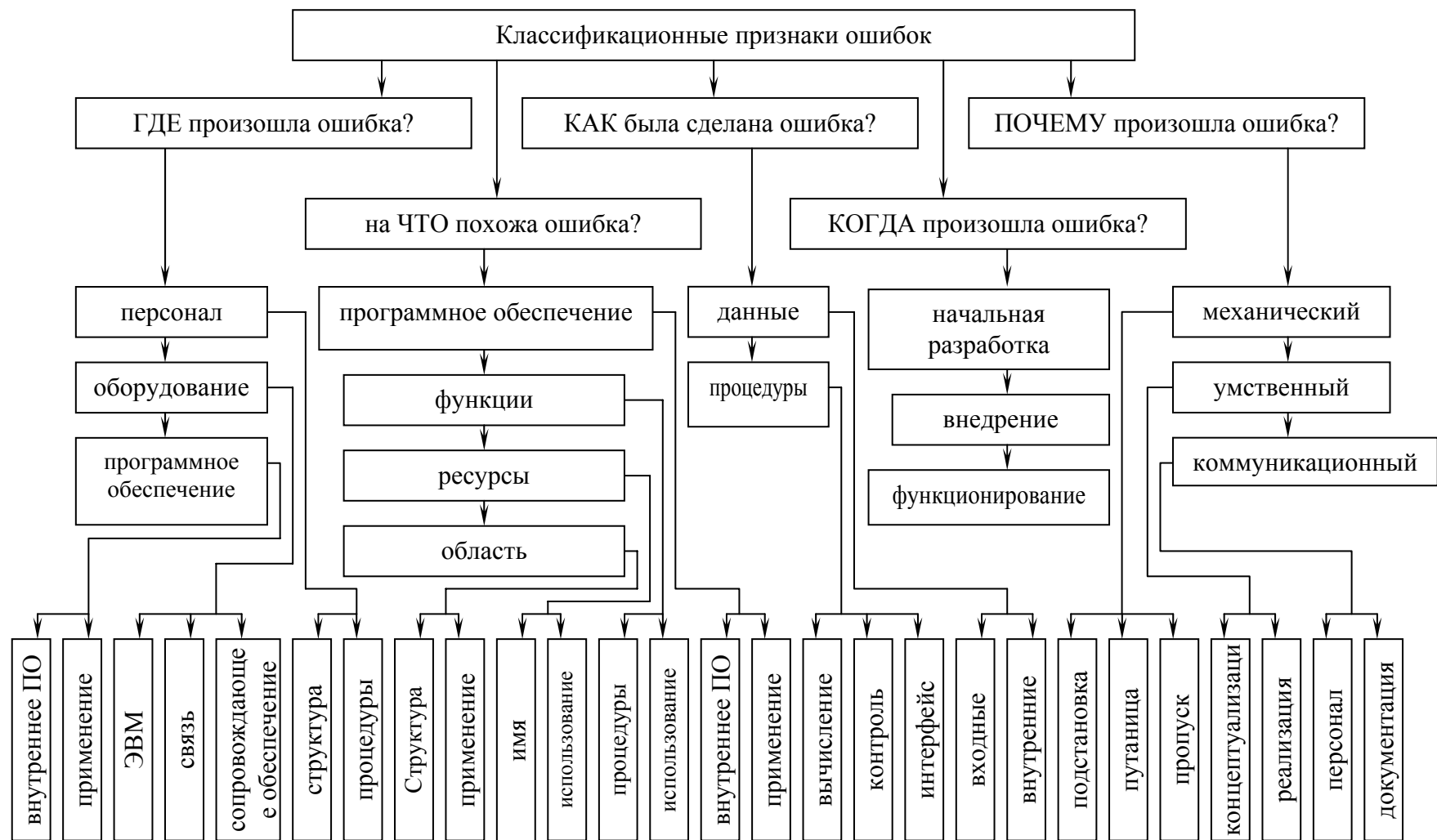
10. Блок-схема системы верификации программных модулей



11. Общая схема отладки программы



12. Классификационная схема ошибок



13. Сложность программного обеспечения



Сложность программы для систем реального времени преимущественно определяется допустимым временем отклика, а для информационно-поисковых систем – количеством типов обрабатываемых переменных.

Вычислительная сложность непосредственно связана с ресурсами вычислительной системы, необходимыми для получения совокупности законченных результатов.

Временной сложностью алгоритма называется время счета, затрачиваемое программой для получения результатов на некоторой эталонной ЭВМ, в зависимости от объема исходных данных.

Программная сложность характеризуется длиной программы или объемом памяти ЭВМ, необходимой для размещения программного комплекса.

Информационную сложность можно представить как объем базы данных, обрабатываемых комплексом программ, или как емкость оперативной и внешней памяти, используемой для накопления и хранения информации при исполнении программ.

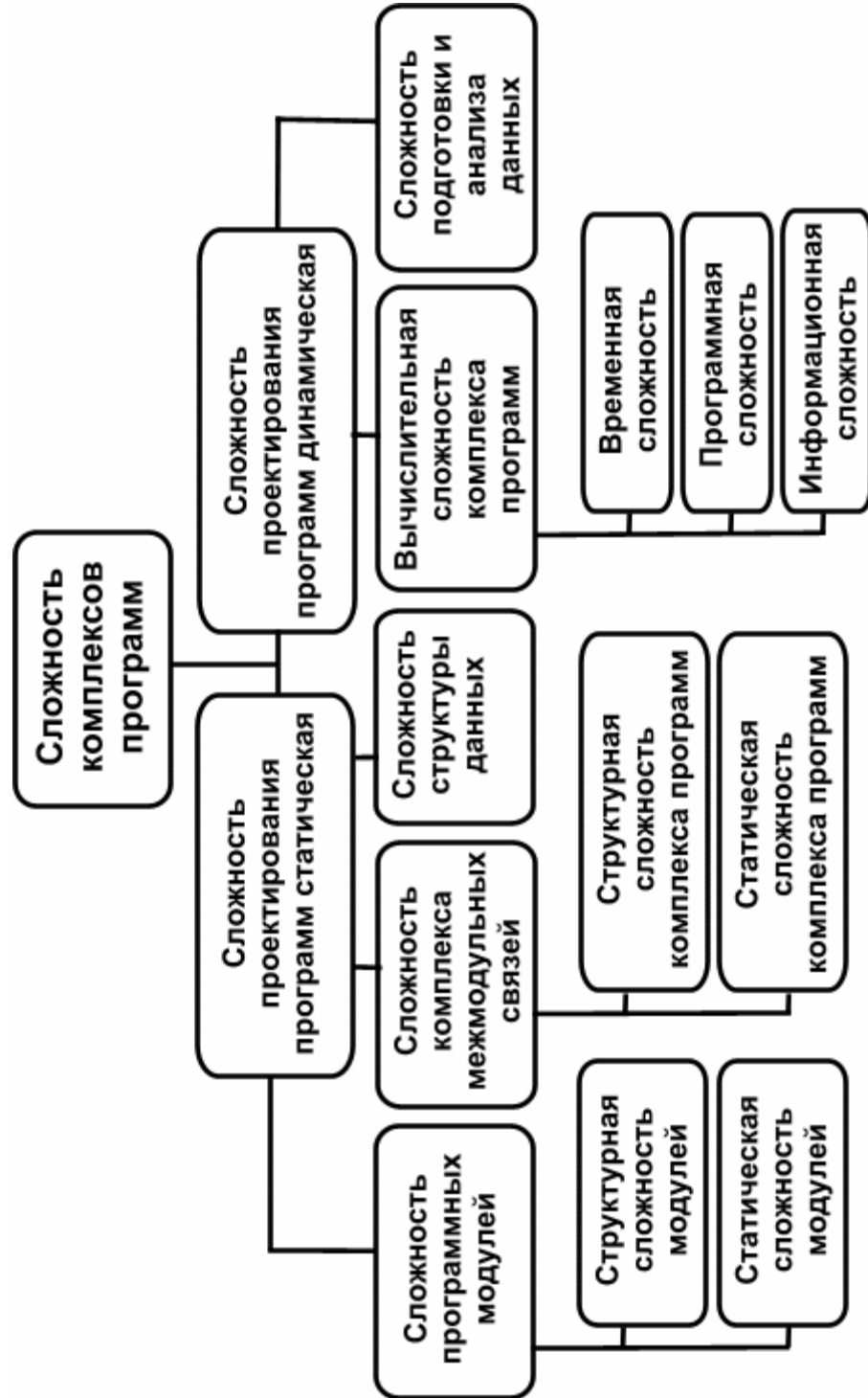
Сложность текста – это длина самого короткого двоичного слова, содержащего всю информацию, необходимую для восстановления рассматриваемого текста при помощи некоторого способа декодирования.

Структурная сложность программ определяется числом взаимодействующих компонент, числом связей между компонентами и сложностью их взаимодействия.

Сложность некоторой межмодульной связи в процессе проектирования можно характеризовать вероятностью ошибки при ее формализации и степенью влияния этой ошибки на последующее функционирование модулей.



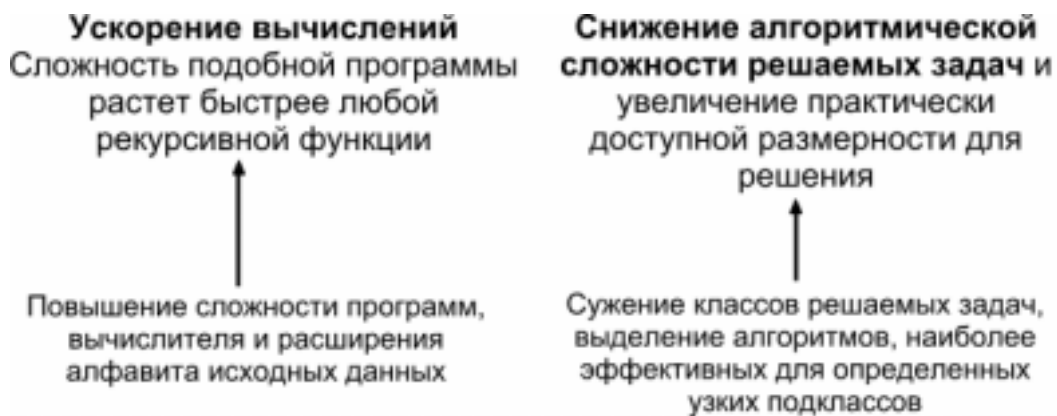
14. Основные виды сложности



15. Схема взаимодействия показателей вычислительной сложности и основные факторы, влияющие на их значения



Теорией сложности установлены явления «сжимания» и «ускорения»:



16. Определение характеристик сложности

Связи	Для модуля	Для ПС (многомодульная программа)
1. Узел	Точка ветвления модуля	Модуль, имеющий более одного выхода
2. Дуга	Последовательные участки модуля	Последовательность нескольких модулей, имеющих один выход
3. Петля	Циклические участки модуля	Циклические участки, состоящие из нескольких модулей

17. Надежность программных средств



Вероятность безотказной работы – это вероятность того, что в пределах заданной наработки отказ системы не возникает.

Вероятность отказа – вероятность того, что в пределах заданной наработки отказ системы возникает.

Интенсивность отказов системы – это условная плотность вероятности возникновения отказа ПС в определенный момент времени при условии, что до этого времени отказ не возник.

Средняя наработка до отказа – математическое ожидание времени работы ПС до очередного отказа.

Среднее время восстановления – математическое ожидание времени восстановления.

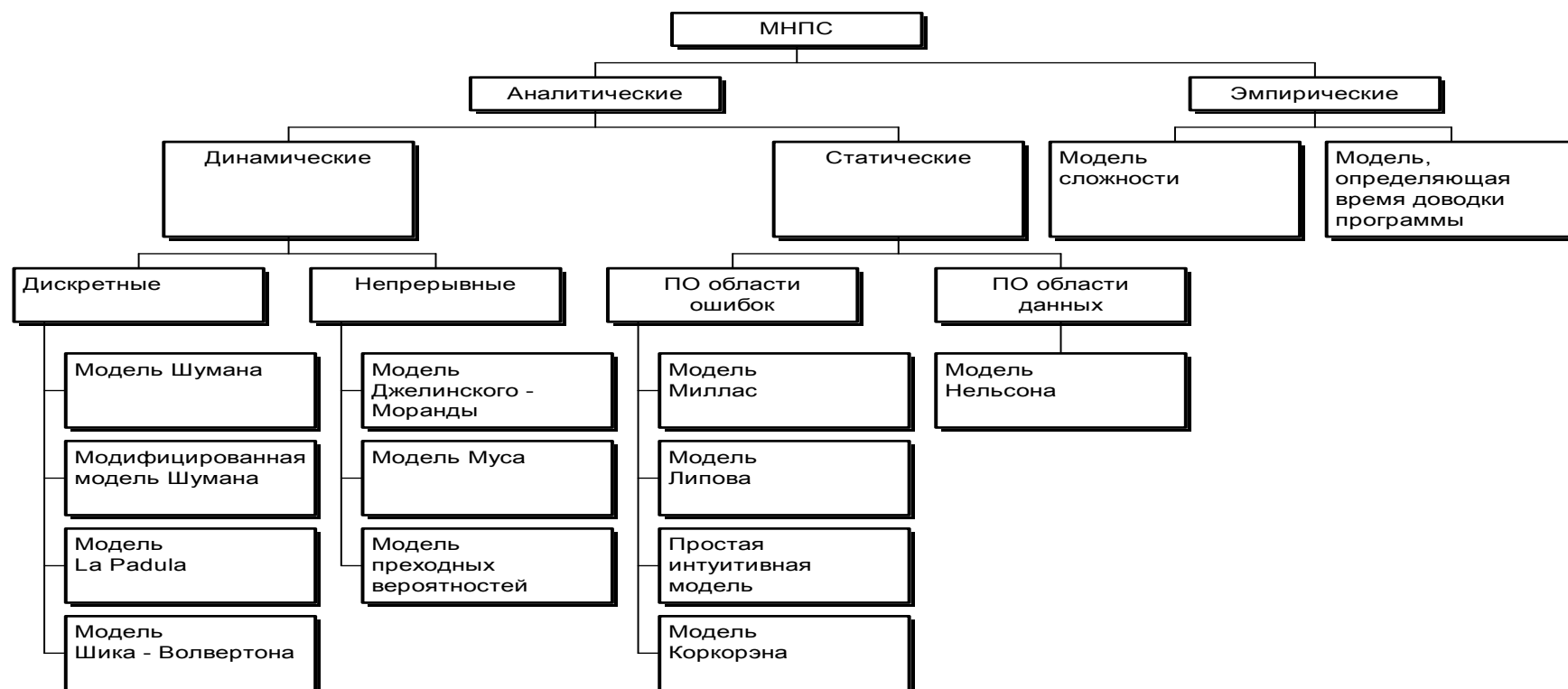
Коэффициент готовности – вероятность того, что ПС ожидается в работоспособном состоянии в произвольный момент времени его использования по назначению.

18. Сбор данных, необходимых для расчета матрицы вероятностей Р

Описание ошибки	Каким модулем вызвана ошибка	Действия на другие модули
-----------------	------------------------------	---------------------------



Классификация моделей надежности ПС.



19. Эффективность



Эффективность – это мера соотношения затрат и результатов функционирования программного средства.

Экономический эффект – это результат внедрения некоторого мероприятия, выраженный в стоимостной форме в виде экономии от его осуществления.

Коэффициент эффективности капитальных вложений – это величина годового прироста прибыли, образующегося в результате производства или эксплуатации программного средства на 1 рубль единоразовых капитальных вложений.

Срок окупаемости – это величина, обратная коэффициенту эффективности. Представляет собой период времени, в течение которого затраты на программное средство окупятся полученным эффектом.

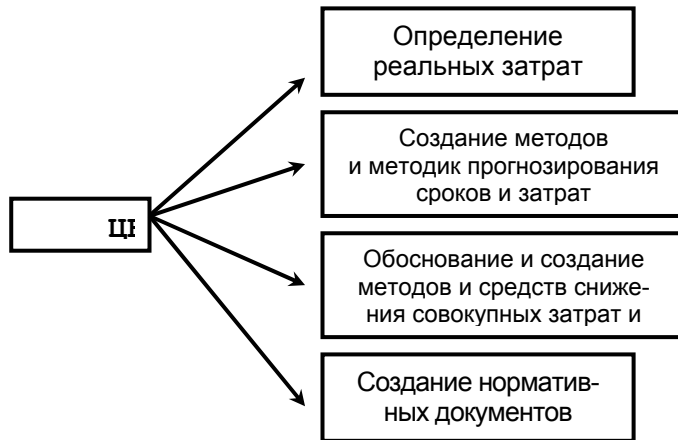
В процессе разработки программного средства в качестве критериев экономической эффективности могут быть выбраны следующие критерии:

1. максимальная экономическая эффективность функционирования программного средства за весь период жизненного цикла при ограниченных затратах на разработку программ;
2. минимальные затраты на разработку программ при заданной экономической эффективности применения и заданном качестве программного средства;
3. максимальное отношение экономической эффективности применения программного средства в течение времени эксплуатации к затратам на его создание;
4. максимальная разность эффекта от функционирования программного средства за весь жизненный цикл и затрат на его разработку, эксплуатацию и сопровождение.



20. Разработка программных средств

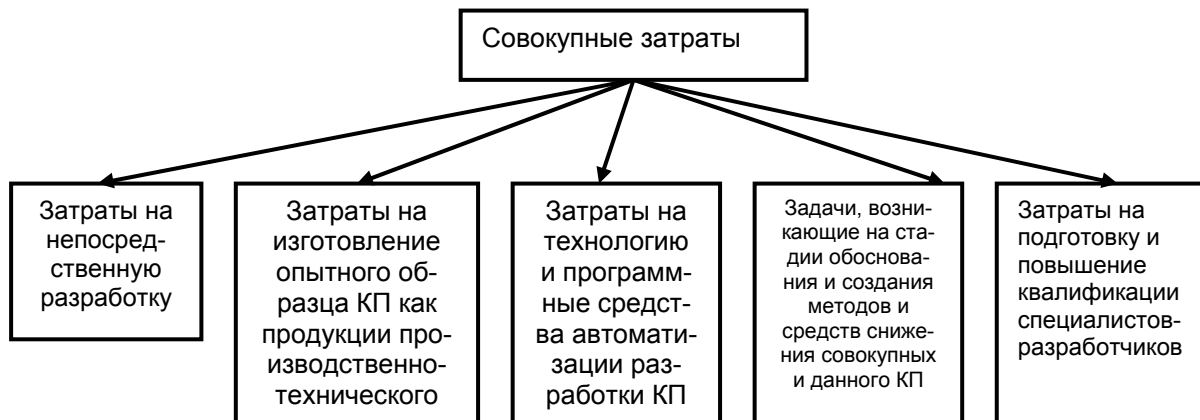
Цели технико-экономического анализа разработки



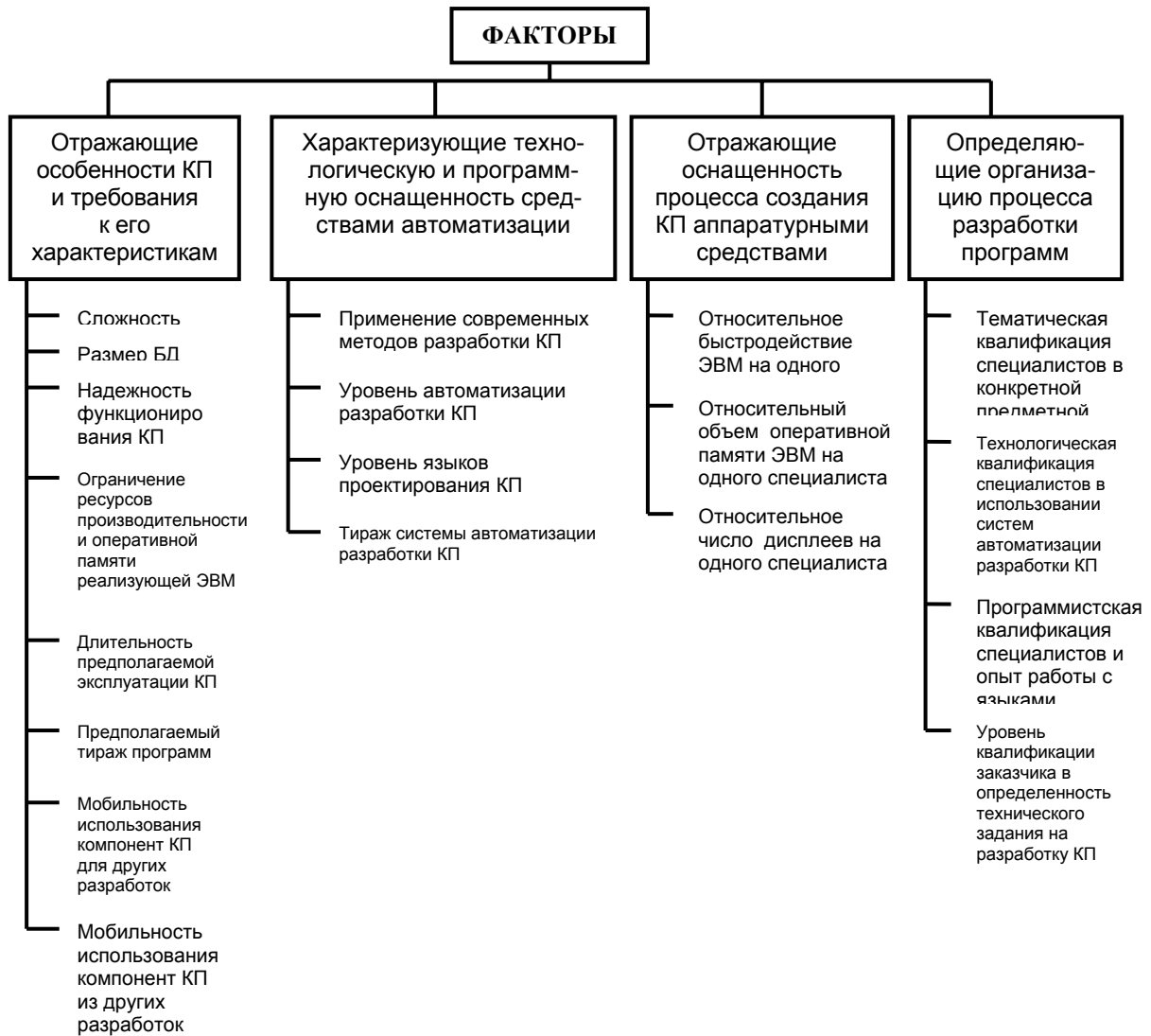
Задачи, возникающие на стадии обоснования и создания методов и средств снижения совокупных затрат и сроков разработки КП



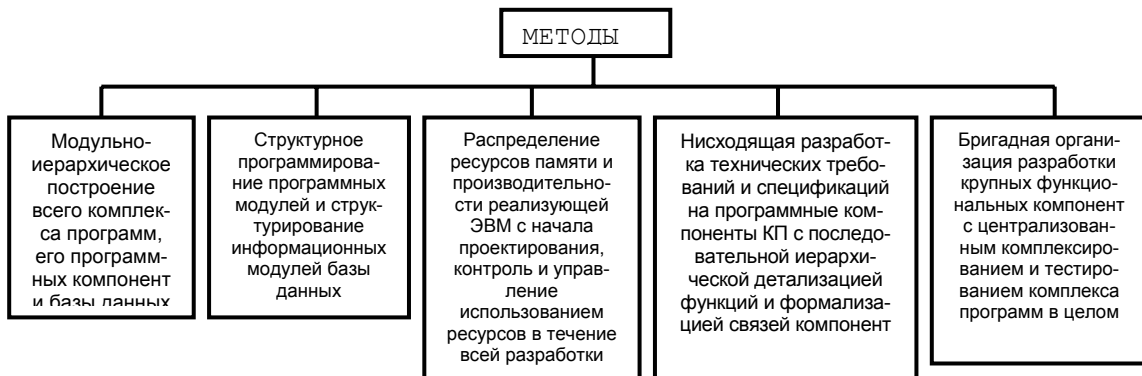
Составляющие затрат на разработку программ



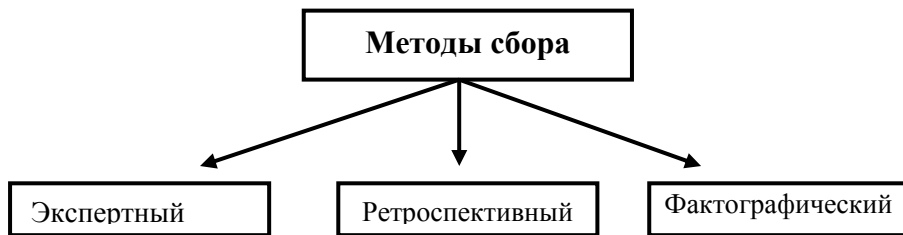
21. Факторы, определяющие затраты на создание ПС



Методы разработки КП



Методы сбора технико-экономической информации о разработках ПС



Основные затраты, снижающие идеальную экономическую эффективность ПС



22. Сертификация



Под **сертификацией** понимается действие третьей стороны, доказывающее, что обеспечивается необходимая уверенность в том, что должным образом идентифицированная продукция, процесс или услуга соответствует конкретному стандарту или другому нормативному документу.

Сертификация может быть **обязательной** или **добровольной**.

Решение о выдаче сертификата на ПС основывается на оценке степени его соответствия действующим и/или специально разработанным документам.



Сертификация

Обязательная

Программные средства, выполняющие особо ответственные функции, в которых недостаточное качество, ошибки или отказы могут нанести большой ущерб или опасны для жизни и здоровья людей (авиация, атомная энергетика, системы управления органами власти, банковские системы...)

Добровольная

Для удостоверения качества ПС с целью повышения их конкурентоспособности, расширения сферы использования и получения дополнительных экономических преимуществ. Таким сертификационным испытаниям подвергаются компоненты операционных систем и ППП широкого применения, повышение гарантий качества которых выгодно как для поставщиков, так и для пользователей ПС.

Решение о выдаче сертификата на ПС основывается на оценке степени его соответствия действующим и/или специально разработанным **документам:**

Действующие международные и национальные стандарты на тестирование, испытания, аттестацию программ и БД.

Международные и государственные стандарты на технологию создания компонент ПС и алгоязыки

Стандарты на сопровождающую ПС документацию

Технические условия, описания, спецификации и другие эксплуатационные документы по выбору

Исходные данные для сертификационных испытаний

Критерии и четко определенные значения показателей качества, которые должны быть достигнуты для выдачи в последующем сертификата соответствия

Значения исходных и результирующих данных, в пределах которых должны удовлетворяться заданные показатели качества

Стандарты, нормативные документы, методики точных воспроизводимых измерений показателей качества, состав и значение исходных и результатных данных

Организационная структура системы сертификации



В процессе испытаний должны проверяться и корректироваться инструкции по эксплуатации комплекса программ в следующих режимах:

- **Генерация пользовательской версии ПС и установка ее на аппаратуре пользователя;**
- контроль работоспособности программ и функциональный контроль всего ПС перед включением рабочего режима;
- нормальное рабочее функционирование всех программ в условиях и ограничениях, заданных в документации;
- аварийные и критические (стрессовые) ситуации, при которых должна сохраняться работоспособность программ;
- диагностика компонент программ и аппаратуры, поиска неисправностей или источника искажений;
- профилактические работы, контроль носителей информации и программ, их дублирование и т.д.



Обязанности специалистов-сертификаторов

Обеспечение полноты и объективности проведения испытаний, достоверности и точности их результатов

Соблюдение порядка и сроков проведения испытаний, согласованных с заявителем, а также условий, обеспечивающих конфиденциальность их проведения

Предотвращение распространения сертифицированного продукта с нарушениями порядка, установленного законодательством, заказчиком или разработчиком

Сохранение государственных и фирменных секретов согласно требованиям действующих нормативных документов

Обеспечение соответствия технического состояния контрольно-измерительной аппаратуры требованиям эксплуатационной документации



Практикум по дисциплине

Практические занятия

Практические занятия проводятся в учебных группах с целью закрепления пройденного теоретического материала, излагаемого в лекциях. Практические занятия по каждой теме проводятся в соответствии с планом распределения времени.

Материал, освоенный на практических занятиях, позволяет выполнить лабораторную работу.

Контроль знаний студентов на практических занятиях проверяется путем проведения аудиторных контрольных работ, а также опроса студентов, показывающего степень усвоения ими материала.

Самостоятельная работа студентов заключается в изучении методов и средств, обеспечивающих оценку качества и надежности ПС, моделирования требуемого уровня надежности ПС с заданными критериями.

Лабораторная работа

№ п/п	Содержание работы	Объем в часах	
		аудиторн. занятия	самост. работа
1	Оценка качественных и технико-экономических показателей программных средств	2	8

1. Задачи по теме «Оценка надёжности ПО»

Модель Миллса.

Задача 1.

Предположим в программе 3 собственных ошибки, внесём ещё 6 случайным образом. В процессе тестирования было найдено 5 ошибок из рассеянных и 2 собственные.

Найти надёжность по модели Миллса.

Задача 2.

Предположим в программе 2 собственных ошибки, внесём ещё 3 случайным образом. В процессе тестирования было найдено 2 ошибки из рассеянных и 3 собственные.

Найти надёжность по модели Миллса.

Задача 3.

Предположим в программе 10 собственных ошибки, внесём ещё 5 случайным образом. В процессе тестирования было найдено 8 ошибок из рассеянных и 3 собственные.

Найти надёжность по модели Миллса.

Задача 4.

Предположим в программе 12 собственных ошибки, внесём ещё 6 случайным образом. В процессе тестирования было найдено 7 ошибок из рассеянных и 5 собственные.

Найти надёжность по модели Миллса.

Задача 5.

Предположим в программе перед началом тестирования было 90 ошибок. Искусственно рассеяно 20 ошибок. В процессе тестовых прогонов было выявлено следующее количество ошибок:

N	25	20	20	15	10	9	1
V	5	4	2	4	2	2	1

Оценить число возможно оставшихся ошибок в программе после каждого тестового прогона. Меру доверия к модели оценить после последнего прогона (С – ?). Построить график зависимости возможного числа ошибок от № прогона.

Задача 6.

Предположим в программе перед началом тестирования было 100 ошибок. Искусственно рассеяно 15 ошибок. В процессе тестовых прогонов было выявлено следующее количество ошибок:

N	20	20	25	16	9	8	1
V	5	3	3	4	3	1	1

Оценить число возможно оставшихся ошибок в программе после каждого тестового прогона. Меру доверия к модели оценить после последнего прогона (С – ?). Построить график зависимости возможного числа ошибок от № прогона.

Задача 7.

Предположим в программе перед началом тестирования было 80 ошибок. Искусственно рассеяно 30 ошибок. В процессе тестовых прогонов было выявлено следующее количество ошибок:

N	30	15	15	13	10	10	1
V	5	3	3	4	3	1	1

Оценить число возможно оставшихся ошибок в программе после каждого тестового прогона. Меру доверия к модели оценить после последнего прогона (С – ?). Построить график зависимости возможного числа ошибок от № прогона.

Простая интуитивная модель.**Задача 1.**

В процессе тестирования программы 1-я группа нашла 15 ошибок, 2-я группа нашла 25 ошибок, общих ошибок было 5. Определить надёжность по простой интуитивной модели.

Задача 2.

В процессе тестирования программы 1-я группа нашла 10 ошибок, 2-я группа нашла 20 ошибок, общих ошибок было 8. Определить надёжность по простой интуитивной модели.

Задача 3.

В процессе тестирования программы 1-я группа нашла 20 ошибок, 2-я группа нашла 22 ошибки, общих ошибок было 4. Определить надёжность по простой интуитивной модели.

Задача 4.

В процессе тестирования программы 1-я группа нашла 5 ошибок, 2-я группа нашла 2 ошибки, общих ошибок было 2. Определить надёжность по простой интуитивной модели.

Задача 5.

В процессе тестирования программы 1-я группа нашла 35 ошибок, 2-я группа нашла 25 ошибки, общих ошибок было 20. Определить надёжность по простой интуитивной модели.

Задача 6.

В процессе тестирования программы 1-я группа нашла 5 ошибок, 2-я группа нашла 40 ошибок, общих ошибок было 5. Определить надёжность по простой интуитивной модели.

Модель Коркорэна.**Задача 1.**

Оттестировать и оценить надёжность по модели Коркорэна. Было проведено 100 испытаний программы. 20 из 100 испытаний прошли безуспешно, а в остальных случаях получились следующие данные:

Тип ошибки	Вероятность появления	Вероятность появления ош. при исп. N_i .
1.Ошибки вычисления	0,09	5
2.Логические ошибки	0,26	25
3.Ошибки ввода/вывода	0,16	3
4.Ошибки манипулирования данными	0,18	—
5.Ошибки сопряжения	0,17	11
6.Ошибки определения данных	0,08	3
7.Ошибки в БД	0,06	4

Задача 2.

Оттестировать и оценить надёжность по модели Коркорэна. Было проведено 100 испытаний программы. 20 из 100 испытаний прошли безуспешно, а в остальных случаях получились следующие данные:

Тип ошибки	Вероятность появления	Вероятность появления ош. при исп. N_i .
1.Ошибки вычисления	0,26	5
2.Логические ошибки	0,9	—
3.Ошибки ввода/вывода	0,8	4
4.Ошибки манипулирования данными	0,2	25
5.Ошибки сопряжения	0,17	11
6.Ошибки определения данных	0,08	3
7.Ошибки в БД.	0,16	3

Задача 3.

Оттестировать и оценить надёжность по модели Коркорэна. Было проведено 100 испытаний программы. 20 из 100 испытаний прошли безуспешно, а в остальных случаях получились следующие данные:

Тип ошибки	Вероятность появления	Вероятность появления ош. при исп. N_i .
1. Ошибки вычисления	0,09	8
2. Логические ошибки	0,26	–
3. Ошибки ввода/вывода	0,17	4
4. Ошибки манипулирования данными	0,2	11
5. Ошибки сопряжения	0,8	25
6. Ошибки определения данных	0,08	3
7. Ошибки в БД	0,16	5

Задача 4.

Оттестировать и оценить надёжность по модели Коркорэна. Было проведено 100 испытаний программы. 20 из 100 испытаний прошли безуспешно, а в остальных случаях получились следующие данные:

Тип ошибки	Вероятность появления	Вероятность появления ош. при исп. N_i .
1. Ошибки вычисления	0,2	4
2. Логические ошибки	0,26	3
3. Ошибки ввода/вывода	0,17	11
4. Ошибки манипулирования данными	0,9	–
5. Ошибки сопряжения	0,08	3
6. Ошибки определения данных	0,8	5
7. Ошибки в БД	0,16	25

Модель Шумана

Задача 1.

Оценить надёжность по модели Шумана.

Дано:

Общее число операторов: 10000

Оценка осуществляется после 10 прогонов.

Данные:

$T_{\text{час}}$	0.5	0.4	0.5	0.75	0.2	0.5	0.3	0.3	0.1	0.4
Кол-во ошибок	2	0	5	3	4	1	3	2	0	1

Задача 2.

Оценить надёжность по модели Шумана.

Дано:

Общее число операторов:9000

Оценка осуществляется после 10 прогонов.

Данные:

T _{час}	0.4	0.5	0.5	0.2	0.75	0.3	0.5	0.3	0.1	0.4
Кол-во ошибок	2	5	0	3	4	1	2	1	1	0

Задача 3.

Оценить надёжность по модели Шумана.

Дано:

Общее число операторов:10000

Оценка осуществляется после 9 прогонов.

Данные:

T _{час}	0.5	0.1	0.3	0.2	0.75	0.3	0.4	0.5	0.5
Кол-во ошибок	1	5	4	1	0	1	2	3	2

Задача 4.

Оценить надёжность по модели Шумана.

Дано:

Общее число операторов:10000

Оценка осуществляется после 8 прогонов.

Данные:

T _{час}	0.5	0.1	0.75	0.5	0.3	0.4	0.2	0.5
Кол-во ошибок	0	5	1	3	1	2	1	2

Задача 5.

Оценить надёжность по модели Шумана.

Дано:

Общее число операторов:19000

Оценка осуществляется после 8 прогонов.

Данные:

T _{час}	0.5	0.3	0.75	0.1	0.3	0.4	0.3	0.4
Кол-во ошибок	1	4	2	2	1	2	2	1

Модель Муса.

Задача 1.

Программа находится в процессе испытаний 15 часов. При этом было выявлено 30 ошибок. Коэффициент сжатия тестов = 5. Первоначальное число ошибок в программе – 100. Заданная наработка на отказ – 3. Количество операторов в программе – 1500.

$$V = 10^8; K = 3 \cdot 10^{-7};$$

Найти надёжность по модели Муса?

Задача 2.

Программа находится в процессе испытаний 15 часов. При этом было выявлено 40 ошибок. Коэффициент сжатия тестов = 6. Первоначальное число ошибок в программе – 90. Заданная наработка на отказ – 4. Количество операторов в программе – 1300.

$$V = 10^8; K = 3 \cdot 10^{-7};$$

Найти надёжность по модели Муса?

Задача 3.

Программа находится в процессе испытаний 15 часов. При этом было выявлено 35 ошибок. Коэффициент сжатия тестов = 4. Первоначальное число ошибок в программе – 120. Заданная наработка на отказ – 6. Количество операторов в программе – 1800.

$$V=1000; K= 0,0000005;$$

Найти надёжность по модели Муса?

Задача 4.

Программа находится в процессе испытаний 15 часов. При этом было выявлено 33 ошибок. Коэффициент сжатия тестов = 5. Первоначальное число ошибок в программе – 80. Заданная наработка на отказ – 2. Количество операторов в программе – 2000.

$$V = 10^8; K = 4 \cdot 10^{-7};$$

Найти надёжность по модели Муса?

2. Лабораторная работа и методические указания по её выполнению

Целью выполнения лабораторной работы является закрепление теоретических знаний и практических навыков в умении разрабатывать и оценивать программные средства.

Результатом выполнения лабораторной работы должно быть готовое и оцененное программное средство, реализованное в среде PASCAL.

2.1. Задание на лабораторную работу

Лабораторная работа состоит из двух частей.

Первая часть предполагает разработку программного средства для решения транспортной задачи линейного программирования. Разработанное ПС должно отвечать следующим параметрам: ориентировано на пользователя, практически не знакомого с персональной техникой; должно иметь эффективный пользовательский интерфейс и средства защиты от несанкционированного доступа. Разработанное ПС должно быть протестировано

но. Тестовые наборы данных и результаты, полученные при проведении машинного эксперимента, должны быть описаны.

Вторая часть лабораторной работы содержит исследование проблем, возникающих при оценке разработанного программного средства. Студенты оценивают качественные показатели ПС, количественные показатели надежности ПС, технико-экономические показатели разработки ПС.

Транспортная задача линейного программирования

Транспортная задача является одной из наиболее распространенных задач линейного программирования и находит широкое практическое приложение.

Постановка транспортной задачи. Некоторый однородный продукт, сосредоточенный у m поставщиков A_i в количестве a_i ($i=1..m$) единиц соответственно, необходимо доставить n потребителям B_j в количестве b_j ($j=1..n$) единиц. Известна стоимость c_{ij} перевозки единицы груза от i -го поставщика к j -му потребителю.

Необходимо составить план перевозок, позволяющий вывести все грузы, полностью удовлетворить потребности и имеющий минимальную стоимость.

Обозначим через x_{ij} количество единиц груза, запланированных к перевозке от i -го поставщика к j -му потребителю. Так как от i -го поставщика к j -му потребителю запланировано к перевозке x_{ij} единиц груза, то стоимость перевозки составит $c_{ij}x_{ij}$.

Стоимость всего плана выразится двойной суммой

$$Z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}$$

Систему ограничений получаем из следующих условий задачи:

а) все грузы должны быть перевезены, т.е.

$$\sum_{j=1}^n x_{ij} = a_i$$

$i=1..m$

б) все потребности должны быть удовлетворены, т.е.

$$\sum_{i=1}^m x_{ij} = b_j$$

$j=1..n$

Таким образом, математическая модель транспортной задачи имеет следующий вид:
Найти минимальное значение линейной функции

$$Z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \quad (1.1)$$

При ограничениях

$$\sum_{j=1}^n x_{ij} = a_i$$

$i=1..m$ (1.2)

$$\sum_{i=1}^m x_{ij} = b_j \quad j=1..n \quad (1.3)$$

$$x_{ij} \geq 0 \quad i=1..m, j=1..n \quad (1.4)$$

В рассмотренной модели предполагается, что суммарные запасы равны суммарным потребностям, т.е.

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j \quad (1.5)$$

Транспортная задача, в которой суммарные запасы и потребности совпадают, т.е. выполняется условие (1.5), называется закрытой моделью; в противном случае – открытой. Для открытой модели может быть два случая:

а) суммарные запасы превышают суммарные потребности

$$\sum_{i=1}^m a_i > \sum_{j=1}^n b_j$$

б) суммарные потребности превышают суммарные запасы

$$\sum_{i=1}^m a_i < \sum_{j=1}^n b_j$$

Линейная функция одинакова в обоих случаях, изменяется только вид системы ограничений.

Найти минимальное значение линейной функции

$$Z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}$$

При ограничениях

$$\sum_{j=1}^n x_{ij} \leq a_i$$

$i=1..m$ (случай 'а')

$$\sum_{i=1}^m x_{ij} = b_j$$

$j=1..n$

$x_{ij} \geq 0$

$$\sum_{j=1}^n x_{ij} = a_i \quad i=1..m \quad (\text{случай 'б'})$$

$$\sum_{i=1}^m x_{ij} \leq b_j \quad j=1..n$$

$$x_{ij} \geq 0$$

Открытая модель решается приведением к закрытой модели.

В случае (а), когда суммарные запасы превышают суммарные потребности, вводится фиктивный потребитель V_{n+1} , потребность которого

$$b_{n+1} = \sum_{i=1}^m a_i - \sum_{j=1}^n b_j$$

В случае (б), когда суммарные потребности превышают суммарные запасы, вводится фиктивный поставщик A_{m+1} , запасы которого

$$a_{m+1} = \sum_{j=1}^n b_j - \sum_{i=1}^m a_i$$

Как стоимость перевозки единицы груза до фиктивного потребителя, так и стоимость перевозки груза от фиктивного поставщика полагаются равными нулю, так как груз в обоих случаях не перевозится.

Транспортная задача имеет $n + m$ уравнений с неизвестными.

Матрицу $X=(x_{ij})_{m,n}$, удовлетворяющую условиям (1.2) – (1.4), называют планом перевозок транспортной задачи (x_{ij} - перевозками).

План X^* , при котором целевая функция (1.1) обращается в минимум, называется оптимальным.

Для разрешимости транспортной задачи необходимо и достаточно, чтобы выполнялось условие баланса

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$$

Опорный план транспортной задачи содержит $m+n-1$ положительных перевозок.

Решение транспортной задачи осуществляется в два этапа: на первом составляют первоначальный опорный план, на втором – оптимизируют его.

Методы составления первоначального опорного плана:

1. Северо-западного угла.
2. Юго-восточного угла.
3. Минимального элемента в строке.
4. Минимального элемента в столбце.

5. Минимального элемента в матрице.
6. Двойного предпочтения.
7. Фогеля.
8. Лебедева.
9. Лебедева – Тихомирова.

Методы оптимизации:

1. Потенциалов.
2. Квадратов.
3. Распределительный.
4. Венгерский.
5. Форда- Фулкерсона.
6. Разрешающих слагаемых.
7. Дифференциальных рент.

Студент, по желанию, реализуя транспортную задачу, выбирает метод построения опорного плана и его оптимизации.

2.2. Методические рекомендации по тестированию программного средства

Тестирование является одним из этапов жизненного цикла ПС, направленным на повышение качественных характеристик.

Программы, как объекты тестирования, имеют ряд особенностей, которые отличают процесс их тестирования от общепринятого, применяемого при разработке аппаратуры и других технических изделий. Особенности тестирования программных средств являются:

- отсутствие эталона (программы), которому должна соответствовать тестируемая программа;
- сложность программ и принципиальная невозможность исчерпывающего тестирования;
- практическая невозможность создания единой методики тестирования (формализация процесса тестирования) в силу большого разнообразия программных средств по их сложности, функциональному назначению, области использования и т.д.

Применительно к программному средству тестирование – процесс многократного выполнения программ с целью обнаружения ошибок.

Цель тестирования – выявление как можно большего количества ошибок. Тестовый прогон считается удачным, если он позволяет выявить ошибки; эффективным, если имеет высокую вероятность обнаружения большего числа ошибок.

Существуют следующие методы тестирования программ:

- *статический*; наиболее формализованный, базируется на правилах структурного построения программ и обработки данных. Проверка степени выполнения этих правил проводится без изменения объектного кода программы путем формального анализа текста программы на языке программирования. Операторы и операнды текста программы анализируются в символьном виде, поэтому этот метод тестирования иногда называют *символьным* тестированием;
- *детерминированный*; наиболее трудоемкий и детализированный метод тестирования, требует многократного выполнения программы с использованием определенных, специальным образом подобранных тестовых наборов данных. При детерминированном тестировании контролируются каждая комбинация исходных данных и соответствующие

результаты, а также каждое утверждение в спецификации тестируемой программы. Детерминированное тестирование в силу трудоемкости, возможно, применять для отдельных модулей в процессе сборки программы или для небольших и несложных программных комплексов;

При тестировании программного средства невозможно перебрать все комбинации исходных данных и проконтролировать результаты функционирования на каждой из них, поэтому для комплексного тестирования программного средства применяется *стохастическое* тестирование. Оно предполагает использование в качестве исходных данных множества случайных величин с соответствующими распределениями. Стохастическое тестирование применяется в основном для обнаружения ошибок;

Программное средство, предназначенное для работы в системах реального времени, должно проходить *тестирование в реальном масштабе времени*.

Каждый из рассмотренных методов тестирования не исключает последовательного применения другого метода, требование к повышению качества программного средства предполагает необходимость подвергать его различным методам тестирования и (или) их сочетаниям в зависимости от сложности области применения.

Для тестирования программного средства лабораторной работы рекомендуется детерминированное и стохастическое тестирование, и их сочетание.

Детерминированное тестирование или тестирование на определенных входных данных, основывается на двух подходах: структурное тестирование и функциональное тестирование. Структурное тестирование, или тестирование программ как «белого ящика» (стратегия тестирования, управляемого логикой программы), предполагает детальное изучение текста (логики) программы и построение таких входных данных, которые позволили бы при многократном выполнении программы обеспечить выполнение максимально возможного количества маршрутов, логических ветвлений, циклов и так далее. Функциональное тестирование, или тестирование программ, как «Черного ящика» (тестирование по «входу – выходу»), полностью абстрагируется от логики программы, предполагается, что логика программы неизвестна, а тестовые наборы подбираются на основании анализа функциональных входных спецификаций.

При построении наборов данных по принципу «белого ящика» руководствуются следующими критериями:

- покрытие операторов; предполагает выбор такого тестового набора данных, который вызывает выполнение каждого оператора в программе хотя бы один раз. Очень слабый критерий.

- покрытие узлов ветвления; предполагает разработку такого количества тестов, чтобы в каждом узле ветвления был обеспечен переход по веткам «истина» и «ложь» хотя бы один раз.

- покрытие условий; если узел ветвления содержит более одного условия, тогда нужно разработать число тестов, достаточное для того, чтобы возможные результаты каждого условия выполнялись, по крайней мере, один раз, каждой точке входа в программу должно быть передано управление при вызове, по крайней мере, один раз.

- комбинаторное покрытие условий; используется для выполнения ошибок в логических выражениях. Требуется создания такого числа тестов, чтобы все возможные комбинации результатов условия в каждом решении и все точки входа выполнялись, по крайней мере, один раз.

К стратегии «черного ящика» относятся методы:

- эквивалентного разбиения. Осуществляется в два этапа – выделение классов эквивалентности, построение тестов. Классом эквивалентности называется множество вход-

ных значений, каждое из которых имеет одинаковую вероятность обнаружения конкретного типа ошибки.

➤ анализ граничных значений. Предполагает исследование ситуаций, возникающих на границах и вблизи границ эквивалентных разбиений.

➤ метод функциональных диаграмм. Заключается в преобразовании входной спецификации программы в функциональную диаграмму (диаграмму причинно-следственных связей) с помощью простейших булевских отношений, построение таблицы решений (методом обратной трассировки), которая является основой для написания эффективных тестовых наборов данных.

1. В спецификации программы выделяются причины и следствия. Причина – отдельное входное условие или класс эквивалентности входных условий. Следствие – выходное условие или результат преобразования системы. Каждой причине и последствию приписывается уникальный номер.

2. Анализируется семантическое содержание спецификации, которая преобразуется в булевский граф, связывающий причины и следствия. Каждая вершина может находиться в состоянии «истина» или «ложь».

3. Диаграмма снабжается примечаниями, задающими ограничения и описывающими комбинации причин и (или) следствий, которые являются невозможными из-за синтаксических или внешних ограничений.

4. По полученной функциональной диаграмме строится таблица решений. Для этого поочередно для каждого следствия, значение которое условно устанавливается в «истину», прослеживается обратный путь (по диаграмме) ко всем причинам, связанным с этим следствием, и фиксируется их состояние. Каждый столбец таблицы решений соответствует тесту.

5. Столбцы решений преобразуют в тесты.

В лабораторной работе выполнить тестирование программного средства, предварительно выбрав и обосновав выбранные методы тестирования.

2.3. Методические рекомендации по оценке программного средства

В лабораторной работе оцениваются качественные показатели ПС, показатели надежности ПС, технико-экономические показатели ПС.

1. Оценка качественных показателей программного средства.

Методика оценки качественных показателей ПС основана на составлении метрики ПС.

В лабораторной работе необходимо выполнить следующее:

1. Выбрать показатели качества (не менее 10) и сформулировать их сущность. Каждый показатель должен быть существенным, т.е. должны быть ясны потенциальные выгоды его использования. Показатели представить в виде таблицы (таблица 1).

Таблица 1

Показатели качества	Сущность показателя	Экспертная оценка (вес) w_i	Оценка, установленная экспериментом r_i

2. Установить веса показателей w_i ($\sum w_i = 1$).

3. Для каждого показателя установить конкретную численную оценку r_i от 0 до 1, исходя из следующего:

- 0 – свойство в ПС присутствует, но качество его неприемлемо;
- 0.5 – 1 – свойство в ПС присутствует и обладает приемлемым качеством;
- 1 – свойство в ПС присутствует и обладает очень высоким качеством.

Возможно присвоение промежуточных значений в соответствии с мнением оценивающего лица относительно полезности того или иного свойства ПС.

4. Определить качество ПС как иерархическую взвешенную сумму весов отдельных показателей. Качество показателя = $w_i * r_i$.

$$ПК = \frac{\sum w_i \cdot r_i}{\text{общее количество показателей}} \quad (1.6)$$

5. Определить среднее значение оценки качества ПС.

6. Представить выходные данные:

- **перечень всех показателей с оценкой 0 с указанием причин такой оценки;**
- **гистограмму, показывающую распределение показателей по интервалам оценок;**
- **какие дефекты ПС обнаружены в результате анализа показателей качества.**

Оценка надежности программного средства.

Одной из важнейших характеристик качества программного средства является надежность.

Надежность – свойство программного средства сохранять работоспособность в течение определенного периода времени, в определенных условиях эксплуатации с учетом последствий для пользователя каждого отказа.

Работоспособным называется такое состояние программного средства, при котором оно способно выполнять заданные функции с параметрами, установленными требованиями технического задания. С переходом в неработоспособное состояние связано событие отказа.

Причиной отказа программного средства является невозможность его полной проверки в процессе тестирования и испытаний. При эксплуатации программного средства в реальных условиях может возникнуть такая комбинация входных данных, которая вызовет отказ, следовательно, работоспособность программного средства зависит от входных данных, и чем меньше эта зависимость, тем выше уровень надежности.

Для оценки надежности используются три группы показателей: качественные, порядковые и количественные.

К основным количественным показателям надежности программного средства относятся:

➤ Вероятность безотказной работы $P(t_3)$ – это вероятность того, что в пределах заданной наработки отказ системы не возникает. Нарботка – продолжительность или объем работ.

$$P(t_3) = P(t \geq t_3),$$

где t – случайное время работы ПС до отказа, t_3 – заданная наработка.

➤ Вероятность отказа – вероятность того, что в пределах заданной наработки отказ системы возникает. Этот показатель, обратный предыдущему.

$$Q(t_3) = 1 - P(t_3),$$

➤ Интенсивность отказов системы $\lambda(t)$ – это условная плотность вероятности возникновения отказа программного средства в определенный момент времени при условии, что до этого времени отказ не возник.

$$\lambda(t) = f(t) / P(t),$$

где $f(t)$ – плотность вероятности отказа в момент времени t .

$$f(t) = \frac{dQ(t)}{dt} = \frac{d}{dt}[1 - P(t)] = -\frac{d}{dt}P(t).$$

Существует следующая связь между $\lambda(t)$ и $P(t)$:

$$P(t) = \exp(-\int \lambda(t)dt)$$

В частном случае $\lambda = \text{const}$.

$$P(t) = \exp(-\lambda_i t).$$

Если в процессе тестирования фиксируется число отказов за определенный интервал времени, то $\lambda(t)$ – число отказов в единицу времени.

➤ Средняя наработка до отказа T_i – математическое ожидание времени работы программного средства до очередного отказа

$$T_i = \int t f(t) dt,$$

где t – время работы программного средства от $(K-1)$ до K отказа.

$$T_i = (t_1 + t_2 + \dots + t_n) / n,$$

где t_i – время работы программного средства между отказами, n – количество отказов.

➤ Среднее время восстановления T_v – математическое ожидание времени восстановления $t_{вi}$ – времени, затраченного на восстановление и локализацию отказа – $t_{о.л.i}$, времени устранения отказа – $t_{у.о.i}$, времени пропускной проверки работоспособности – $t_{п.п.i}$.

$$t_{вi} = t_{о.л.i} + t_{у.о.i} + t_{п.п.i}.$$

Для этого показателя термин «время» означает время, затраченное программистом на перечисленные виды работ.

➤ Коэффициент готовности K_2 – вероятность того, что программное средство ожидается в работоспособном состоянии в произвольный момент времени его использования по назначению:

$$K_2 = T_i / (T_i + T_v).$$

Причиной отказа программного средства являются ошибки, которые могут быть вызваны: внутренним свойством программного средства, реакцией программного средства на изменение внешней среды функционирования. Это значит, что при самом тщательном тестировании, если предположить, что удалось избавиться от всех внутренних ошибок, нельзя с полной уверенностью утверждать, что в процессе эксплуатации программного средства не возникнет отказ.

Основным средством определения количественных показателей надежности являются модели надежности, под которыми понимают математическую модель, построенную для оценки зависимости надежности от заранее известных или оцененных в ходе создания программного средства параметров. В связи с этим определение надежности показателей принято рассматривать в единстве трех процессов – предсказание, измерение, оценивание.

Предсказание – это определение количественных показателей надежности исходя из характеристик будущего программного средства.

Измерение – это определение количественных показателей надежности, основанное на анализе данных об интервалах между отказами, полученных при выполнении программ в условиях тестовых испытаний.

Оценивание – это определение количественных показателей надежности, основанное на данных об интервалах между отказами, полученными при испытании программного средства в реальных условиях функционирования.

Все модели надежности можно классифицировать по тому, какой из перечисленных процессов они поддерживают (предсказывающие, прогнозные, оценивающие, измеряющие). Нужно отметить, что модели надежности, которые в качестве исходной информации используют данные об интервалах между отказами, можно отнести к измеряющим, и к оценивающим в равной степени. Некоторые модели, основанные на информации, полученной в ходе тестирования программного средства дают возможность делать прогнозы поведения программного средства в процессе эксплуатации.

Рассмотрим аналитические и эмпирические модели надежности.

Аналитические модели дают возможность рассчитать количественные показатели надежности, основываясь на данных о поведении программы в процессе тестирования (измеряющие и оценивающие модели). Эмпирические модели базируются на анализе структурных особенностей программ. Они рассматривают зависимость показателей надежности от числа межмодульных связей, количества циклов в модулях, отношения количества прямолинейных участков к количеству точек ветвления и тому подобное. Нужно отметить, что часто эмпирические модели не дают конечных результатов показателей надежности.

Аналитическое моделирование надежности программного средства включает четыре шага:

- определение предложений, связанных с процедурой тестирования программного средства;
- разработка или выбор аналитической модели, базирующейся на предположениях о процедуре тестирования;
- выбор параметров моделей с использованием полученных данных;
- применение модели – расчет количественных показателей надежности по модели.

Аналитические модели представлены двумя группами: динамические и статические модели. В динамических моделях надежности программного средства поведение программы (появление отказов) рассматривается во времени. В статических моделях появление отказов не связывают со временем, а учитывают только зависимость количества ошибок от числа тестовых прогонов (по области ошибок) или зависимость количества ошибок от характеристики входных данных (по области данных). Для использования ди-

намических моделей необходимо иметь данные о появлении отказов во времени. Статические модели принципиально отличаются от динамических тем, что в них не учитывается время появления ошибок в процессе тестирования и не используется никаких предположений о поведении функции риска $\lambda(t)$. Эти модели строятся на твердом статистическом фундаменте.

Модель Коркорэна.

Применение модели предполагает знание следующих ее показателей:

- модель содержит изменяющуюся вероятность отказов для различных источников ошибок и соответственно разную вероятность их исправления;
- в модели используются такие параметры, как результат только N испытаний, в которых наблюдается N_i ошибок i -го типа;
- выявление в ходе N испытаний ошибки i -го типа появляется с вероятностью a_i .

Показатель уровня надежности R вычисляют по следующей формуле:

$$R = \frac{N_0}{N} + \sum_{i=1}^k \frac{Y_i * (N_i - 1)}{N}, \quad (1.7)$$

где

N_0 – число безотказных (или безуспешных) испытаний, выполненных в серии из N испытаний,

k – известное число типов ошибок,

Y_i – вероятность появления ошибок,

при $N_i > 0$, $Y_i = a_i$,

при $N_i = 0$, $Y_i = 0$.

Модель Шумана.

Модель Шумана относится к динамическим моделям дискретного времени, данные для которой собираются в процессе тестирования программного обеспечения в течение фиксированных или случайных интервалов времени. Модель Шумана предполагает, что тестирование проводится в несколько этапов. Каждый этап представляет собой выполнение программы на полном комплексе разработанных тестовых данных. Выявленные ошибки регистрируются, но не исправляются. В конце этапа рассчитываются количественные показатели надежности, исправляются найденные ошибки, корректируются тестовые наборы и проводится следующий этап тестирования. В модели Шумана предполагается, что число ошибок в программе постоянно и в процессе корректировки новые ошибки не вносятся. Скорость обнаружения ошибок пропорциональна числу оставшихся ошибок.

Предполагается, что до начала тестирования имеется Et ошибок. В течение времени тестирования τ обнаруживается ε_c ошибок в расчете на одну команду в машинном языке.

Таким образом, удельное число ошибок на одну машинную команду, оставшихся в системе после τ времени тестирования, равно:

$$\varepsilon_r(\tau) = Et / It * \varepsilon_c(\tau), \quad (1.8)$$

где It – общее число машинных команд, которое предполагается постоянным в рамках этапа тестирования.

Предполагается, что значение функции частоты отказов $Z(t)$ пропорционально числу ошибок, оставшихся в программе после израсходованного на тестирование времени t .

$$Z(t) = C * \varepsilon_r(\tau)$$

где C – некоторая постоянная
 t – время работы программы без отказов.

Тогда, если время работы программы без отказа t отсчитывается от точки $t = 0$, а τ остается фиксированным, функция надежности, или вероятность безотказной работы на интервале от 0 до t , равна

$$R(t, \tau) = \exp \{-C * [Et / It - \varepsilon_c(\tau)] * t\} \quad (1.9)$$

$$t_{cp} = 1 / \{C * [Et / It - \varepsilon_c(\tau)]\}. \quad (1.10)$$

Нам необходимо найти начальное значение ошибок Et и коэффициент пропорциональности – C . В процессе тестирования собирается информация о времени и количестве ошибок на каждом прогоне, т.е. общее время тестирования τ складывается из времени каждого прогона

$$\tau = \tau_1 + \tau_2 + \tau_3 + \dots + \tau_n.$$

Предполагая, что интенсивность появления ошибок постоянна и равна λ , можно вычислить ее как число ошибок в единицу времени

$$\lambda = \frac{\sum_{i=1}^k A_i}{\tau}, \quad (1.11)$$

где A_i – количество ошибок на i – ом прогоне.

Имея данные для двух различных моментов тестирования τ_a и τ_b , которые выбираются произвольно с учетом требования, чтобы $\varepsilon_c(\tau_b) > \varepsilon_c(\tau_a)$, можно сопоставить уравнения (1.10) и (1.12) при τ_a и τ_b .

$$t_{cp} = \frac{\tau}{\sum_{i=1}^k A_i}. \quad (1.12)$$

$$\frac{1}{\lambda_{\tau_a}} = \frac{1}{C[E_t / It - \varepsilon_c(\tau_a)]}; \quad (1.13)$$

$$\frac{1}{\lambda_{\tau_b}} = \frac{1}{C[E_t / It - \varepsilon_c(\tau_b)]}. \quad (1.14)$$

Неизвестный параметр C (1.16) получается путем подстановки Et (1.15) в выражение (1.13). Вычисляя соотношения (1.13) и (1.14) получим

$$E_t = \frac{It[\lambda\tau_b / \lambda\tau_a \varepsilon_c(\tau_b) - \varepsilon_c(\tau_b)]}{(\lambda\tau_b / \lambda\tau_a) - 1}. \quad (1.15)$$

$$C = \frac{\lambda\tau_a}{[E_t / I_t - \varepsilon_c(\tau_a)]}. \quad (1.16)$$

Получив неизвестные Et и C , можно рассчитать надежность программы по формуле (1.9), проведем расчеты применительно к учебной программе.

Например, в программе имеется $I_t = 4381$ оператор. В процессе последовательных тестовых прогонов были получены следующие данные:

№ прогона	1	2	3	4	5	6	7	8	9	10
Кол-во ошибок	1	2	1	1	1	1	1	2	1	1
Время (м)	5	8	2	1	5	1	1	2	5	5

Выберем две точки, исходя из требования, чтобы число ошибок, найденных на интервале $A \div B$, и было больше, чем на интервале $0 \div A$. За точку A возьмем 2 прогон, а за точку B – 8 прогон. Тогда ошибки, найденные на этапах тестирования на интервалах $0 \div A$ и $A \div B$, будут равны соответственно:

$$\varepsilon_c(\tau_A) = 3 / 4381 = 0.0007$$

$$\varepsilon_c(\tau_B) = 7 / 4381 = 0.0015.$$

Время тестирования на интервалах равно:

$$\tau_A = 13$$

$$\tau_B = 12.$$

Рассчитаем интенсивности появления ошибок на двух интервалах:

$$\lambda_A = 3 / 13 = 0.23$$

$$\lambda_B = 7 / 12 = 0.58.$$

$$Et = \frac{4381(0.58/0.23 \times 0.0007 - 0.0015)}{0.58/0.23 - 1} = 0.763 \approx 1 \text{ ошибка}.$$

Тогда число имеющихся до начала тестирования ошибок равно

$$C = \frac{0.23}{0.76 / 4381 - 0.0007} = -460.$$

Рассчитаем вероятность безотказной работы в течение времени t при $\tau = 35$ мин.

$$R(t, 35) = \exp\{460 \times [0.763 / 4381 - \varepsilon_c(35)]t\}$$

Возьмем $t = 60$ мин.

$$R(60,35) = \exp\{460 \times [0.0002 - 0.0027]60\} \approx 0.9$$

Таким образом, надежность безотказной работы достаточно велика и вероятность сбоев и возникновения ошибок небольшая.

Оценка технико-экономических показателей разработки программных средств.

В лабораторной работе необходимо определить следующие показатели:

- объём программного средства (в операторах языка или строках текста)-О;
- длительность разработки (по фактическому времени)-Д;

$$Д=Д1-Д2,$$

где Д1- дата начала разработки технического задания на ПС,

Д2- дата сдачи ПС;

- число программных и информационных модулей в ПС -Р;
- количество фактически затраченного времени на разработку ПС- М;
- трудоёмкость разработки ПС (по фактически затраченному времени по стадиям разработки) – Т;

- абсолютное снижение трудовых затрат- ТА

$$ТА= Т0- Т1,$$

где Т0- трудовые затраты на решение транспортной задачи по базовому варианту (вручную, на данных , представленных преподавателем),

Т1- трудовые затраты на решение транспортной задачи по предлагаемому варианту;

- коэффициент относительного снижения трудовых затрат- КТ

$$КТ=ТА/Т0*100,$$

- индекс снижения трудовых затрат или повышение производительности труда- УТ

$$УТ= Т0/Т1,$$

- абсолютное снижение стоимостных затрат- СА

$$СА= С0 – С1,$$

где С0 – стоимостные затраты на решение транспортной задачи по базовому варианту,

С1- стоимостные затраты на решение транспортной задачи по предлагаемому варианту;

- коэффициент относительного снижения стоимостных затрат- КС

$$КС=СА/С0 * 100,$$

- индекс снижения стоимости затрат -УС

$$УС=С0/С1,$$

- срок окупаемости ПС- К

$$К=КТ/СА,$$

где КТ- затраты на разработку и внедрение программного средства.

2.4. Методические рекомендации по обеспечению технологической безопасности программного средства

В целях повышения надежности и устойчивости защиты информации и ПС разрабатываются и используются специальные программные средства защиты:

- А) идентификация пользователей и определение их полномочий;
- Б) идентификация терминалов;
- В) обеспечение защиты файлов;
- Г) защита операционных систем и проблемных программ;
- Д) вспомогательные программы защиты.

В лабораторной работе предлагается реализовать алгоритм установления подлинности пользователя, обеспечивающий законность обращения пользователя к разработанному ПС (рис. 1)

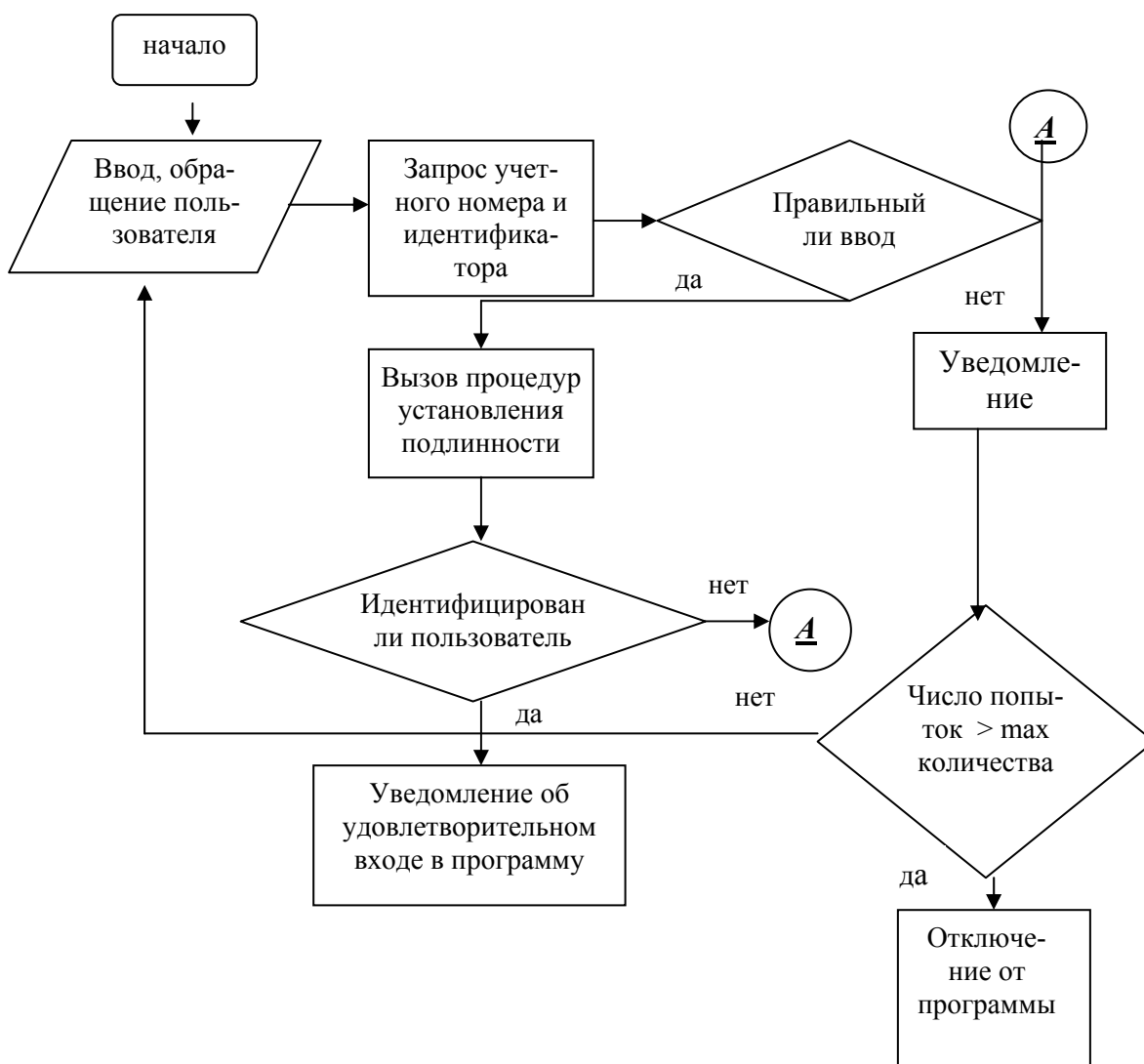


Рис. 1. Алгоритм идентификации установления подлинности пользователя

2.5. Методические указания по оформлению лабораторной работы

Разработанное программное обеспечение должно быть оформлено стандартным образом, и содержать разделы в соответствии с приведенным ниже перечнем.

1. Титульный лист.
2. Содержание.
3. Задание на выполнение лабораторной работы.
4. Описание программы
 - 4.1. Общие сведения
 - 4.2. Логическая структура программного комплекса
 - 4.3. Входные данные
 - 4.4. Выходные данные
 - 4.5. Вызов и загрузка
5. Тестирование программы
 - 5.1. Программа проведения тестирования
 - 5.2. Набор тестовых данных
 - 5.3. Анализ результатов тестирования
6. Оценка разработанной программы
 - 6.1. Выбор и обоснование методов оценки
 - 6.2. Расчет показателей качества
 - 6.3. Расчет показателей надежности
 - 6.4. Оценка технико-экономических показателей
7. Управление технологической безопасностью программы и данных.
8. Список литературы.
9. Приложения
 - 9.1. Распечатки экранов
 - 9.2. Текст программы
 - 9.3. Результаты тестирования и выполнения программы
 - 9.4. Заставка.

Каждый раздел должен содержать только ту информацию, которая требуется. При возникновении дублирования допускается ссылка по данному вопросу на другой раздел отчета.

Раздел «Входные данные» содержит описание входных данных, форму их подготовки для ввода. Раздел не должен содержать конкретных данных выполнения программы.

Раздел «Выходные данные» должен иллюстрировать форму выдачи результатов работы программы. Выходными данными являются, кроме результатов решения задачи, все сообщения программы.

При оценке количественных показателей надежности обязательно:

- Задаться требуемой наработкой на отказ;
- Рассчитать текущую величину наработки на отказ и, если она меньше заданной, определить дополнительное возможное число отказов и время тестирования, необходимое, чтобы достичь требуемого уровня;
- Определить возможное количество оставшихся в программе ошибок;
- Определить уровень надежности разработанной программы;

Заставка к программе должна содержать информацию об авторе программы, компьютере и системе программирования, преподавателе и дате выполнения.

При оформлении лабораторной работы необходимо учитывать:

- Названия параграфов должны соответствовать их наименованию, указанному в содержании;
- Подчеркивание наименований параграфов и др. не допускаются;
- Все страницы работы должны быть пронумерованы последовательно арабскими цифрами и соответствовать проставленным страницам в содержании;
- Лабораторная работа должна быть жестко скреплена:
- Не допускается наличие элементов оформления в карандаше.

Тестовые задания по дисциплине

1. **Наука** об измерениях, методах, средствах обеспечения их единства и способах достижения требуемой точности – это:

1. сертификация;
2. качество ПС;
3. метрология;
4. технология разработки ПС.

2. **Основное понятие** метрологии – это:

1. измерение;
2. полезность;
3. устойчивость;
4. экономическая эффективность.

3. **Предметом** изучения метрологии является:

1. исследование жизненного цикла программных средств;
2. оценка совокупности характеристик объекта, относящихся к его способности удовлетворять установленные и предполагаемые потребности;
3. извлечение количественной информации о свойствах объектов и процессов с заданной точностью и достоверностью.

4. **Объектом** изучения метрологии является:

1. характеристика жизненного цикла программного средства;
2. испытание и сопровождение программного средства;
3. программное средство;
4. тестирование программного средства.

5. **Прообразом ГОСТ** в России, определяющим цели, методы, причины и пути развития метрологии, являлся указ:

1. о пошлинах и сборах от 1900 года;
2. о введении подушной подати от 1723 года;
3. о системе мер и весов от 1841 года;
4. о пересчете душ всея Руси от 1519 года.

6. **Основоположник** метрологии в России

1. Флоренский П.А.;
2. Менделеев Д.И.;
3. Магницкий Л.Ф.;

7. **В метрологии, понятие «измерение»** определяется как:

1. нахождение значения физической величины опытным путем с помощью специальных технических средств;
2. установление степени соответствия данного ПО внешним спецификациям;
3. техническая процедура для определения одной или нескольких специфических характеристик материала или изделия.

8. **Теория единиц физических величин, теория исходных средств измерений и теория передачи размеров единиц физических величин** исследуется в теорииизмерений.

1. построения средств
2. точности
3. единства

9. **Метод испытания** – это:

1. техническая процедура для определения одной или нескольких специфических характеристик материала или изделий;
2. установленные технические правила проведения испытаний;
3. сочетание принципов и средств измерений, соответствующих выбранному принципу.

10. **Государственная эталонная база России** включает:

1. 116 первичных и специальных эталонов;
2. 85 специальных эталонов;
3. 116 первичных эталонов;
4. 85 первичных и специальных эталонов.

11. **Метрология** включает науку:

1. об измерениях;
2. о надежности ПС;
3. о контроле качества;
4. о методах измерений;
5. о средствах обеспечения единства измерений;

12. **Нахождение значения физической величины** опытным путем с помощью специальных технических средств – это:

1. метод измерений;
2. методика испытаний;
3. измерение;
4. метод испытания.

13. **Метод измерения** – это:

1. сочетание принципов и средств измерений;
2. измерение значения физической величины опытным путем с помощью специальных технических средств;
3. техническая процедура измерения одной или нескольких специфических характеристик материала или изделий.

14. **В теорию точности средств измерений** входят:

1. теория единиц физических величин;
2. теория исходных средств измерений;
3. теория погрешности средств измерений;
4. теория средств измерений;
5. средства измерения;
6. теория метрологической надежности средств измерений;
7. методы измерения;
8. принципы и методы нормирования и определения метрологических характеристик – теория методов измерений.

15. **Метрологическая служба** в законе РФ «Об обеспечении единства измерений» определяется как:

1. совокупность операций, выполняемых с целью определения и подтверждения действительных значений метрологических характеристик;
2. деятельность, направленная на проверку средств измерений;

3. совокупность субъектов деятельности и видов работ, направленных на обеспечение единства измерений.

16. Согласно «Закону об обеспечении единства измерений» РФ от 23 апреля 1993 года **«техническое устройство, предназначенное для измерений»** является основным для термина:

1. средство измерений;
2. единство измерений;
3. эталон единицы величины;
4. метрологическая служба.

17. **Теоретическая метрология** изучает:

1. вопросы практического применения в различных сферах;
2. комплексы правил, требований и норм, используемых государством для контроля за единообразием измерений;
3. общие вопросы теории измерений.

18. **Комплексы правил, требований и норм для контроля единообразия измерений** исследуется в разделе метрологии:

1. сертификационном;
2. исследовательском;
3. законодательном;
4. теоретическом;
5. прикладном;

19. **Процесс установления и применения научных и организационных основ технических средств, правил и норм, необходимых для достижения единства и требуемой точности измерений** – это процесс:

1. метрологического обеспечения;
2. метода испытания;
3. измерения;
4. сертификации;
5. оценки качества и надежности ПС.

20. **Законодательная метрология** изучает:

1. комплексы правил, требований и норм, используемых государством для контроля за единообразием измерений;
2. совокупность субъектов деятельности, видов работ и норм, направленных на единство измерений;
3. функциональные критерии и требования, отражающие специфику применения и целевого назначения программных средств.

21. **Общие вопросы теории измерений** изучаются вразделе метрологии:

1. прикладном;
2. классификационном;
3. теоретическом;
4. законодательном.

22. Раздел **«Основные представления метрологии»** включает:

1. основные понятия и термины;
2. методологию измерений;
3. теорию погрешности средств измерений;
4. постулаты метрологии;
5. учение о физических величинах;
6. теорию методов измерений;
7. средства измерения;
8. теорию единиц физических величин.

23. Согласно «Закону об обеспечении единства измерений» РФ понятие **«эталон единицы величины»** – это:

1. состояние измерений, при котором результаты этих измерений выражены в узаконенных единицах величин;
2. средство измерений, предназначенное для воспроизведения и хранения единицы величины;
3. совокупность мер, способов, характеристик и приемов, направленных на определение одной или нескольких специфических характеристик.

24. **Прикладная метрология** изучает вопросы:

1. нормирования и единообразия измерений;
2. практического применения программных средств в различных сферах деятельности;
3. осуществления оценки качества программных средств.

25. **Совокупность операций, выполняемых органами государственной метрологической службы (другими уполномоченными на то органами, организациями) с целью определения и подтверждения соответствия средства измерений установленным техническим требованиям называют средств измерений.**

1. калибровкой
2. аккредитацией
3. лицензированием ;
4. поверкой.

26. **Метрологическое обеспечение** – это:

1. нахождение значения физической величины опытным путем с помощью специальных технических средств;
2. установление и применение научных и организационных основ технических средств, правил и норм, необходимых для достижения единства и требуемой точности измерений;
3. комплекс нормативных документов, включающих в себя государственные стандарты и другие нормативные документы, определяющие порядок передачи размера величин, порядок проведения испытаний, проверки и калибровки средств измерений.

27. Вопросы практического применения в различных сферах изучает **раздел метрологии:**

1. прикладной;
2. теоретический;
3. исследовательский.

28. К разделу «**Основные представления о метрологии**» относятся:

1. теория методов измерений;
2. теория метрологической надежности средств измерений;
3. теория передачи размеров единиц физических величин;
4. основные понятия и термины;
5. методология измерений.

29. **Калибровка средств измерений** – это:

1. совокупность операций, выполняемых органами государственной метрологической службы (другими уполномоченными на то органами, организациями) с целью определения и подтверждения соответствия средства измерений установленным техническим требованиям;
2. совокупность операций, выполняемых с целью определения и подтверждения действительных значений метрологических характеристик и (или) пригодности к применению средства измерений, не подлежащему государственному метрологическому контролю и надзору;
3. деятельность, осуществляемая органом государственной метрологической службы или метрологической службой юридического лица в целях проверки соблюдения установленных метрологических правил и норм.

30. **Государственные стандарты, применяемые в установленном порядке, международные (региональные) стандарты, правила, положения, инструкции и рекомендации** относятся к:

1. нормативным документам по обеспечению единства измерений;
2. метрологическому контролю и надзору;
3. эталонам средств измерений;
4. метрологическим службам, направленным на обеспечение единства измерений.

31. **Программное средство** – это программа

1. Написанная на любом алгоритмическом языке
2. Укомплектованная спецификацией
3. Укомплектованная руководством пользователя
4. Записанная на техническом носителе, изготовленная по определённой технологии и снабженная эксплуатационной документацией.

32. **ПСВТ** (программное средство вычислительной техники) должно отвечать следующим требованиям:

1. Соответствовать техническим условиям, согласованным с государственным комитетом науки и техники.
2. Отвечать стандартам операционной системы, под которую оно (ПС) написано.
3. ПС должно быть зарегистрировано в государственном фонде алгоритмов и программ.
4. Вводится в эксплуатацию, модернизироваться, и поддерживаться после выпуска.
5. Должно быть объектно-ориентированным и интерактивным.
6. Иметь программную документацию, оформленную в соответствии с ГОСТ РФ.

33. **Качество ПС** – это

1. Показатель, характеризующий объем ресурсов, требуемый для нормального функционирования ПС.

2. Показатель, характеризующий универсальность работы ПС на системах разной конфигурации.
 3. Совокупность характеристик ПС, относящихся к его способности удовлетворять потребности заказчика.
34. **Показатель качества ПС** – это
1. Количественная характеристика одного или нескольких свойств ПС, составляющих его качество, рассматриваемое применительно к условиям создания, эксплуатации и потребления.
 2. Количественная характеристика, показывающая быстроту работы ПС
 3. Количественная характеристика, показывающая насколько эффективно ПС работает в экстремальных условиях.
35. **ПС как объекты** характеризуются:
1. Областью применения ПС, назначение ПС в народном хозяйстве.
 2. Типом решаемых задач с определенной областью применения, степенью связи решаемых задач с масштабом времени или с допустимой длительностью ожидания результатов решения.
 3. Объемом и сложностью совокупности программ, решающую единую задачу, необходимым составом и требуемыми значениями характеристик качества функционирования программ и величиной допустимого ущерба из-за недостаточного качества.
 4. Прогнозируемыми значениями длительности эксплуатации и возможности развития множества версий программ, предполагаемым тиражом производства и применения программ.
 5. Степенью необходимой документируемости ПС.
36. **Функциональные критерии качества ПС** отражают:
1. Специфику областей применения ПС.
 2. Степень соответствия ПС их основному целевому назначению.
 3. Количество разнообразных функций, которые выполняет программа.
 4. Все выше приведенные ответы.
37. **Конструктивные критерии качества ПС** отражают:
1. Эффективность использования программами ресурсов вычислительных средств.
 2. Количество подключаемых к программе модулей.
 3. Надежность функционирования ПС.
 4. Сложность взаимодействия между модулями.
38. **Критерии этапа проектирования** – это
1. Сложность проектирования.
 2. Корректность
 3. Трудоемкость разработки
 4. Все вышеприведенные ответы верны.
39. **Критерии этапа эксплуатации** – это
1. Функциональная сложность и надежность.
 2. Эффективность используемых ресурсов.
 3. Объем исходных и результирующих данных.
 4. Ни один из выше приведенных ответов не отражает в полной мере.

40. **Критерии этапа сопровождения** – это
1. Модернизируемость и сложность ПС.
 2. Мобильность и удаленность от пользователя.
 3. Модернизируемость, мобильность.
 4. Трудоемкость модификации программ.
41. **К функциональной пригодности ПС** относится:
1. Корректность работы.
 2. Разнообразие выполняемых функций.
 3. Защищенность от сбоев и несанкционированных действий. Способность к взаимодействию.
 4. Функциональная корректность, способность к взаимодействию, мобильность.
42. **Критерии удобства использования ПС:**
1. Понятность и обучаемость.
 2. Простота использования.
 3. Комфортная эксплуатация.
 4. Все выше перечисленные ответы.
43. **Виды метрик и шкал для измерения показателей качества ПС** бывают:
1. Интервальные.
 2. Порядковые.
 4. Числовые.
 5. Номинальные.
44. **Функциональные показатели качества БД** – это:
1. Полнота накопленных описаний объекта.
 2. Организация взаимосвязи элементов
 3. Тип БД и его программная реализация
 4. Достоверность и идентичность данных
 5. Актуальность данных
45. **Конструктивные показатели качества БД** – это:
1. Объем и оперативность БД.
 2. Динамичность
 3. Тип БД и его программная реализация
 4. Глубина ретроспективы
 5. Периодичность.
46. **Порядковая шкала** измерения показателей качества ПС
1. Характеризуется относительными величинами или реально измеряемыми физическими показателями.
 2. Позволяет ранжировать некоторые характеристики путём сравнения с опорными значениями.
 3. Характеризует только наличие рассматриваемого признака у объекта, без учёта градации по нему.
47. **Интервальная шкала** измерения показателей качества ПС
1. Характеризуется относительными величинами или реально измеряемыми физическими показателями.

2. Позволяет ранжировать некоторые характеристики путём сравнения с опорными значениями.
 3. Характеризует только наличие рассматриваемого признака у объекта, без учёта градации по нему.
48. **Номинальная шкала** измерения показателей качества ПС
1. Характеризуется относительными величинами или реально измеряемыми физическими показателями.
 2. Позволяет ранжировать некоторые характеристики путём сравнения с опорными значениями.
 3. Характеризует только наличие рассматриваемого признака у объекта, без учёта градации по нему.
49. **Характеристика**, отражающая возможность внесения изменений в ПС:
1. Модифицируемость
 2. Информативность
 3. Структурированность
50. **Характеристика**, обеспечивающая понятность ПС – это:
1. Открытость
 2. Информативность
 3. Структурированность
51. **Длина программы** при учете сложности ПС измеряется
1. В операторах
 2. В страницах
 3. В символах
52. **Группы показателей сложности ПС** – это сложность
1. Проектирования программ
 2. Функционирования программ
 3. Взаимосвязи модулей
 4. Структуры данных
 5. Обработки данных
53. **Сложность проектирования ПО** оценивается на основе сложности
1. программных модулей
 3. структуры
 4. взаимосвязи модулей
 5. структуры данных
 6. обработки данных
54. **К показателям статической сложности ПС** относятся:
1. Сложность программных модулей
 2. Вычислительная сложность комплекса программ
 3. Сложность комплекса и межмодульных связей
 4. Сложность подготовки анализа данных
 5. Сложность структуры данных

55. **К показателям динамической сложности ПС** относятся:
1. Вычислительная сложность комплекса программ
 2. Сложность программных модулей
 3. Сложность структуры данных
 4. Сложность подготовки и анализа данных
56. **Временная сложность ПС** отражает длительность
1. проектирования ПС
 2. обработки данных для получения результатов
 3. установки ПС на отдельный компьютер
57. **Программная сложность ПС** характеризуется
1. Объемом памяти для размещения программы
 2. Сложностью кода ПС
 3. Количеством программистов, разрабатывающих ПС
58. **Информационная сложность ПС** отражает
1. Объем обрабатываемых программой данных
 2. Объем внешней и оперативной памяти для накопления и хранения информации
 3. Оба ответа верны
59. **Сложность связей по управлению** определяется
1. Числом вариантов возможных вызовов данного модуля
 2. Числом модулей, вызываемых из данного
 3. Числом модулей во всей программе
 4. Глобальными характеристиками всех модулей
60. **Сложность связей по информации** определяется
1. Структурой передаваемой в модуль информации
 2. Объемом передаваемой в модуль информации
 3. Подсчетом обменных и глобальных переменных модулей
61. **Корректность ПС** – это
1. Соответствие проверяемого объекта некоторому эталонному объекту или совокупности формализованных характеристик и правил.
 2. Правильность работы ПС под разными операционными системами
 3. Соответствие описаниям в спецификации ПС
 4. Правильность работы ПС при различных входных данных
62. **Корректность тестов программ** включает корректность
1. Синтаксическую
 2. Семантическую
 3. Функциональную
 4. Структурную
 5. Конкретных значений
 6. Межмодульных связей
63. **Корректность программных модулей** включает корректность
1. Синтаксическую
 2. Семантическую
 3. Функциональную

4. Структурную
5. Конкретных значений
6. Межмодульных связей

64. Составляющие корректности данных:

1. Синтаксическая
2. Семантическая
3. Функциональная
4. Структурная
5. Конкретных значений
6. Межмодульных связей

65. Составляющие корректности групп и комплексов программ

1. Синтаксическая
2. Семантическая
3. Функциональная
4. Структурная
5. Конкретных значений
6. Межмодульных связей

66. Компоненты корректности ПС – это корректность

1. Исходных текстов программ
2. Программных модулей
3. Данных
4. Групп и комплексов программ
5. Всё вышеперечисленное

67. Функциональная корректность программных модулей оценивается

1. Детерминированно
2. Стохастически
3. Оба варианта верны

68. Функциональная корректность данных связана

1. С корректностью функционирования данных
2. С конкретизацией данных в процессе исполнения программ
3. Со сложностью тестируемого модуля.

69. Структурная корректность модулей заключается в соответствии структуры модулей

1. Правилам структурного программирования
2. Правилам оформления и внутреннего построения
3. Оба варианта верны

70. Корректность программных модулей – это

1. Соответствие структуры модулей правилам модульного программирования и конкретным правилам построения и оформления
2. Правильность работы каждого отдельного модуля
3. Соответствие модуля правилам ООП

71. **Корректностью обработки исходных данных и полученных результатов** определяется
1. Функциональная корректность
 2. Структурная корректность
 3. Корректность конкретных значений
 4. Динамическая корректность программных модулей
72. **Высказывание** «корректность определяется правилами структурирования и упорядочивания» относится к корректности
1. программных модулей
 2. данных
 3. групп и комплексов программ
 4. тестов программ
73. **Правилами модульного программирования** определяется
1. Правильность работы ПС
 2. Сложность проектирования групп программ
 3. Корректность групп программ
 4. Общие правила написания программ
74. **Основные виды тестирования ПС** – это
1. Стохастический
 2. Динамический
 3. Статический
 4. Детерминированный
75. **Проверку корректности** включает ... вид тестирования
1. Детерминированный
 2. Стохастический
 3. Динамический
76. **В стохастическое тестирование** входит
1. Тестирование полноты исходных данных
 2. Тестирование полноты выполнения функций
 3. Планирование тестирования
 4. Тестирование корректности сравнения с эталоном
 5. Тестирование результатов
77. **Программный модуль**
1. Реализует одну или несколько функций
 2. Всегда отвечает правилам ООП
 3. Имеет определенную логическую структуру
 4. Должен быть независим от внешнего состояния программы
 5. Должен быть минимальным в размерах
 6. Может вызываться в разных контекстах
78. **Для установления корректности программ** необходимы
1. Эталоны
 2. Методы проверки программ

79. **Эталонами** для проверки ПС являются
1. Формализованные правила описания программ, данных, структуры.
 2. Программные спецификации.
 3. Тесты.
 4. Все выше перечисленные типы.
80. **Методы проверки корректности** программ:
1. Проверка соответствия формализованным правилам.
 2. Проверка программных спецификаций.
 3. Тестирование.
81. **Проверка соответствия** формализованным правилам – это:
1. Контроль полноты спецификаций.
 2. Верификация.
 3. Тестирование.
 4. Синтаксический контроль.
 5. Контроль связей модулей по информации и по управлению.
82. **В процессе детерминированного тестирования** определяется:
1. Полнота исходных данных, корректность обработки результатов, корректность сравнения с эталонами.
 2. Полнота выполнения функций, корректность использования ресурсов ЭВМ, надёжность функционирования.
 3. Контроль полноты проверок, корректность корректировок, корректность сравнения с эталонами.
83. **В процессе стохастического тестирования** определяется:
1. Полнота исходных данных, корректность обработки результатов, корректность сравнения с эталонами.
 2. Полнота выполнения функций, корректность использования ресурсов ЭВМ, надёжность функционирования.
 3. Контроль полноты проверок, корректность корректировок, корректность сравнения с эталонами.
84. **В процессе динамического тестирования** определяется:
1. Полнота исходных данных, корректность обработки результатов, корректность сравнения с эталонами.
 2. Полнота выполнения функций, корректность использования ресурсов ЭВМ, надёжность функционирования.
 3. Контроль полноты проверок, корректность корректировок, корректность сравнения с эталонами.
86. **Процесс**, состоящий в проверке и доказательстве корректности программы по отношению к совокупности формальных утверждений, представленных в программной спецификации и полностью определяющих связи между входными и выходными данными этой программы, **называется:**
1. Стандартизацией.
 2. Валидацией.
 3. Верификацией.
 4. Сертификацией.

86. **Инварианты** – это:

1. Условия, не зависящие от входных спецификаций программы и отражающие фактические отношения между переменными и программой.
2. Различные пути следования алгоритма программы в зависимости от входных данных.
3. Условия, зависящие от входных спецификаций программы и отражающие формальные отношения между переменными и программой.

87. **Внутренними дестабилизирующими факторами ПС** являются:

1. Сбои и отказы аппаратуры ЭВМ.
2. Ошибки алгоритмизации задач.
3. Ошибки программирования.
4. Ошибки персонала при эксплуатации.
5. Недостаточное качество средств защиты.
6. Искажение информации в каналах связи.

88. **По оценкам Бейзера** программы, передаваемые на тестирование в среднем на каждые 100 исполняемых операторов содержат ошибок:

1. От 1 до 3
2. От 3 до 5
3. От 5 до 10

89. **Большее количество ошибок** будет обнаружено:

1. При тестировании программы лицом, участвующем в её создании.
2. Не участвующем в её создании.
3. Закономерности не существует.

90. **При тестировании и отладке программы** в первую очередь обнаруживаются ошибки:

1. Первичные.
2. Вторичные.
3. Закономерности не существует.

91. **Свойство программного средства** сохранять работоспособность в течение определенного периода времени в определенных условиях эксплуатации – это

1. надежность
2. корректность
3. устойчивость

92. **Количественные показатели надежности** включают

1. среднее время восстановления после отказа
2. интенсивность отказа
3. коэффициент готовности
4. все вышеперечисленное

93. **Отказом** является переход программного средства из

1. работоспособного состояния в неработоспособное
2. надежного состояния в ненадежное
3. устойчивого состояния в неустойчивое

94. **Коэффициент готовности** – это

1. количество отказов программного средства за единицу времени
2. условная плотность вероятности возникновения отказа программного средства в определенный момент времени
3. вероятность того, что программное средство окажется в работоспособном состоянии в определенный момент времени его использования по назначению

95. **Модели надежности** – это:

1. модели, предназначенные для оценки надежности программ;
2. математические модели, построенные для оценки зависимости надежности программного средства от заранее известных или оцененных в ходе тестирования параметров.
3. физическая модель, определяющая степень надежности программного средства.

96. **Модели надежности** программного средства делятся на:

1. аналитические, эмпирические;
2. конструктивные, статические и организационные;
3. дискретные, непрерывные.

97. **Модель Шумана** относится к ... моделям надежности программных средств

1. аналитическим, динамическим, дискретным;
2. аналитическим, динамическим, непрерывным;
3. статическим, по области ошибок.

98. **Модель Липова** относится к моделям надежности программных средств

1. аналитическим, динамическим, непрерывным;
2. статическим, по области ошибок;
3. аналитическим, динамическим, дискретным;

99. **Модель Нельсона** относится к ... моделям надежности программных средств

1. статическим, по области ошибок;
2. аналитическим, динамическим, дискретным;
3. статическим, по области данных.

100. **Простая интуитивная модель** относится к ... моделям надежности программных средств

1. аналитическим, динамическим, непрерывным;
2. статическим, по области ошибок;
3. аналитическим, динамическим, дискретным;

101. **Модель Миллса** относится к моделям надежности программных средств

1. аналитическим динамическим
2. аналитическим статическим
3. эмпирическим

102. **Модель Коркорэна** относится к ... моделям надежности программных средств

1. аналитическим динамическим
2. аналитическим статическим
3. эмпирическим

103. **Модель Муса** относится к ... моделям надежности программных средств
1. аналитическим динамическим
 2. аналитическим статическим
 3. эмпирическим
104. **Модель сложности** относится к моделям надежности программных средств
1. аналитическим динамическим
 2. аналитическим статическим
 3. эмпирическим
105. **Модель переходных вероятностей** относится к моделям надежности программных средств
1. аналитическим динамическим
 2. аналитическим статическим
 3. эмпирическим
106. **Эффективность ПС** – это:
1. Совокупность характеристик объекта, относящихся к способности удовлетворять некоторые потребности.
 2. Соотношение между результатами деятельности и затратами, произведенными для получения и использования.
 3. Способность решать задачи в короткие сроки.
107. **Основные показатели экономической эффективности ПС** – это:
1. Экономический эффект.
 2. Стоимость.
 3. Коэффициент экономической эффективности капитальных вложений.
 4. Срок окупаемости.
 5. Трудоёмкость.
108. **Экономический эффект ПС** – это:
1. Величина годового прироста прибыли, образующейся в результате производства или эксплуатации ПС на 1 руб. единовременных капитальных вложений.
 2. Результат внедрения ПС, выраженный в стоимостной форме.
 3. Период времени, в течение которого затраты на ПС окупаются полученным эффектом.
109. **Коэффициент эффективности капитальных вложений ПС** – это:
1. Величина годового прироста прибыли, образующейся в результате производства или эксплуатации ПС на 1 руб. единовременных капитальных вложений.
 2. Результат внедрения ПС, выраженный в стоимостной форме.
 3. период времени, в течение которого затраты на ПС окупаются полученным эффектом.
110. **Срок окупаемости ПС** – это:
1. Величина годового прироста прибыли, образующейся в результате производства или эксплуатации ПС на 1 руб. единовременных капитальных вложений.
 2. Результат внедрения ПС, выраженный в стоимостной форме.
 3. Период времени, в течение которого затраты на ПС окупаются полученным эффектом.

111. В число целей технико-экономического анализа разработки ПС не входит:
1. Определение реальных затрат на разработку компонент ПС в целом с учетом их сложности и требуемого качества.
 2. Создание методов и методик прогнозирования сроков и затрат на создание программ.
 3. Минимизация числа сбоев в готовом ПС.
112. Экономический эффект от использования ПС не зависит от
1. Количества ошибок, выявленных в ПС в процессе эксплуатации.
 2. Приведенных затрат на единицу работ, выполненных с помощью базового ПС.
 3. Приведенных затрат на единицу работ, выполненных с помощью нового ПС.
113. Оценка затрат на разработку ПС предполагает оценку:
1. Размер разрабатываемого ПС.
 2. Трудоёмкости разработки.
 3. Продолжительности разработки.
 4. Стоимости разработки.
 5. Все вышеперечисленное
114. Основной мерой оценки размера ПС является количество
1. Операторов.
 2. Операций.
 3. Строк кода.
 4. Функциональных точек.
115. Под функциональной точкой разрабатываемого ПС понимается
1. Входной элемент приложения (входной документ или экранная форма).
 2. Выходной элемент приложения (отчет, документ, экранная форма).
 3. Запрос (вопрос – ответ)
 4. Логический файл.
 5. Интерфейс приложения.
 6. Любой из перечисленных элементов.
116. Факторами, определяющими затраты на создание ПС, являются факторы:
1. Отражающие особенности ПС и требования к его характеристикам.
 2. Характеризующие технологическую и программную оснащенность средствами автоматизации.
 3. Отражающие оснащенность процесса создания ПС аппаратными средствами.
 4. Определяющие организацию процесса разработки программ.
117. Специально упакованные и оформленные для коммерческой продажи, проката, сдачи в аренду или лизинга пакеты программ, разработанные и (или) поставляемые системными или независимыми поставщиками, представляют собой:
1. Программные продукты.
 2. Программные средства.
 3. Системное программное обеспечение.
118. Рыночный жизненный цикл ПС дополнительно включает фазы:
1. Торговый анализ.
 2. Фиксирование маркетинговой стратегии.

3. Тестирование рынка.
4. Коммерциализация.

119. Критерии сегментирования рынка ПС – это:

1. Разбиение по профессиям.
2. Возраст пользователя.
3. Определение статуса национального или международного рынка.

120. Стратегии маркетинга при принятии решения о выходе на рынок с новым типом программного продукта.

1. Массовый маркетинг.
2. Товарно-дифференцируемый маркетинг.
3. Целевой маркетинг.

121. Сертификация – это

1. гарантия качества продукции;
2. показатель работоспособности ПС;
3. независимая оценка, подтверждающая, что ПС удовлетворяет всем требованиям системы стандартов.

122. Для проведения сертификации необходимо:

1. выбрать один из предусмотренных стандартов;
2. выбрать необходимый вид тестирования ПС;
3. изучить алгоритм выполнения ПС.

123. Сертификат – это:

- а) документ, который удостоверяет соответствие системы требованиям стандарта;
- б) документ, содержащий основные положения о сертификации ПС
- в) перечень действующих стандартов.

124. По признаку заинтересованности сторон системы сертификации классифицируются как

- а) национальные;
- б) обязательные;
- в) региональные;
- г) международные.

125. По признаку правового статуса системы сертификации классифицируются на (2 правильных ответа):

- а) самостоятельные;
- б) обязательные;
- в) добровольные;
- г) региональные.

126. Преимущества от сертификации прямо пропорциональны усилиям, вложенным в

1. систему качества;
2. увеличение оптимального внутреннего алгоритма ПС;
3. обеспечение надежности ПС.

127. Проведение обязательной сертификации вызвано:
1. определенными требованиями, налагаемыми на предприятия, в том числе материального характера;
 2. условиями договора между заявителем и органом по сертификации;
 3. желанием пользователя ПС.
128. Сертификация ПС является обязательной:
1. да;
 2. нет;
 3. в зависимости от этапов разработки ПС.
129. Сертификация ПС введена в целях:
1. предоставления определенных гарантий на то, что ПС соответствует всем требованиям нормативных документов;
 2. выявления ошибок в ПС и их устранения;
 3. продажи ПС.
130. Сертификация ПС может быть приостановлена:
1. да;
 2. нет;
 3. нет, если нет соответствующих нормативных документов.
131. Процесс сертификации продолжается 2-3
1. месяца;
 2. года;
 3. дня.
132. Один из законов, в соответствии с которым сертификация проводится в соответствии с законом:
1. «О защите прав труда»;
 2. «О защите прав потребителей»;
 3. «О сертификации».
133. Сертификация приостанавливается или отменяется
1. По желанию заявителя, отмене или замене нормативного документа;
 2. Из-за неактуальности ПС
 3. Из-за непогашенной задолженности по оплате за сертификацию ПС
134. Лаборатория сертификации ПС приобретает необходимые полномочия, если она
1. аттестована
 2. аккредитована
 3. имеет необходимое оборудование
135. Проводится ли сертификация в области метрологии
1. да
 2. нет