Стандартизация разработки программных средств

Выполнила : Никитина А.Р

БИН-21 (оз)

Общие положения о стандартах

- **Стандартизация** – это деятельность, направленная на разработку и установление требований, норм, правил, характеристик, как обязательных для выполнения, так и рекомендуемых, обеспечивающая право потребителя на приобретение товаров надлежащего качества, а так же право на безопасность и комфортность труда.
- **Цель стандартизации** – достижение оптимальной степен упорядочения в той или иной области посредством широкого и многократного использования установленных положений, требований, норм для решения реально существующих, планируемых или потенциальных задач.
- **Объект стандартизации** – продукция, процесс, услугу, для которых разрабатывают те или иные требования, характеристики, параметры, правила и т.п.
- Область стандартизации – совокупность взаимосвязанных объектов стандартизации.

Стандартизация осуществляется на разных уровнях.

- Уровень стандартизации зависит от того, участники какого географического, экономического, политического региона мира принимают стандарт.

- **Региональная стандартизация** – деятельность, открытая только для соответствующих органов государств одного географического, политического или экономического региона.
- **Международная стандартизация** – участие в стандартизации открыто для соответствующих органов любой страны.
- **Национальная стандартизация** – стандартизация в одном конкретном государстве.
- **Административно-территориальная стандартизация** – стандартизация, которая проводится в административно-территориальной единице(провинции, крае и тп.)

Нормативные документы по стандартизации и виды стандартов

В процессе стандартизации вырабатываются нормы, правила, требования, характеристики, касающиеся объекта стандартизации, которые оформляются в виде нормативного документа.

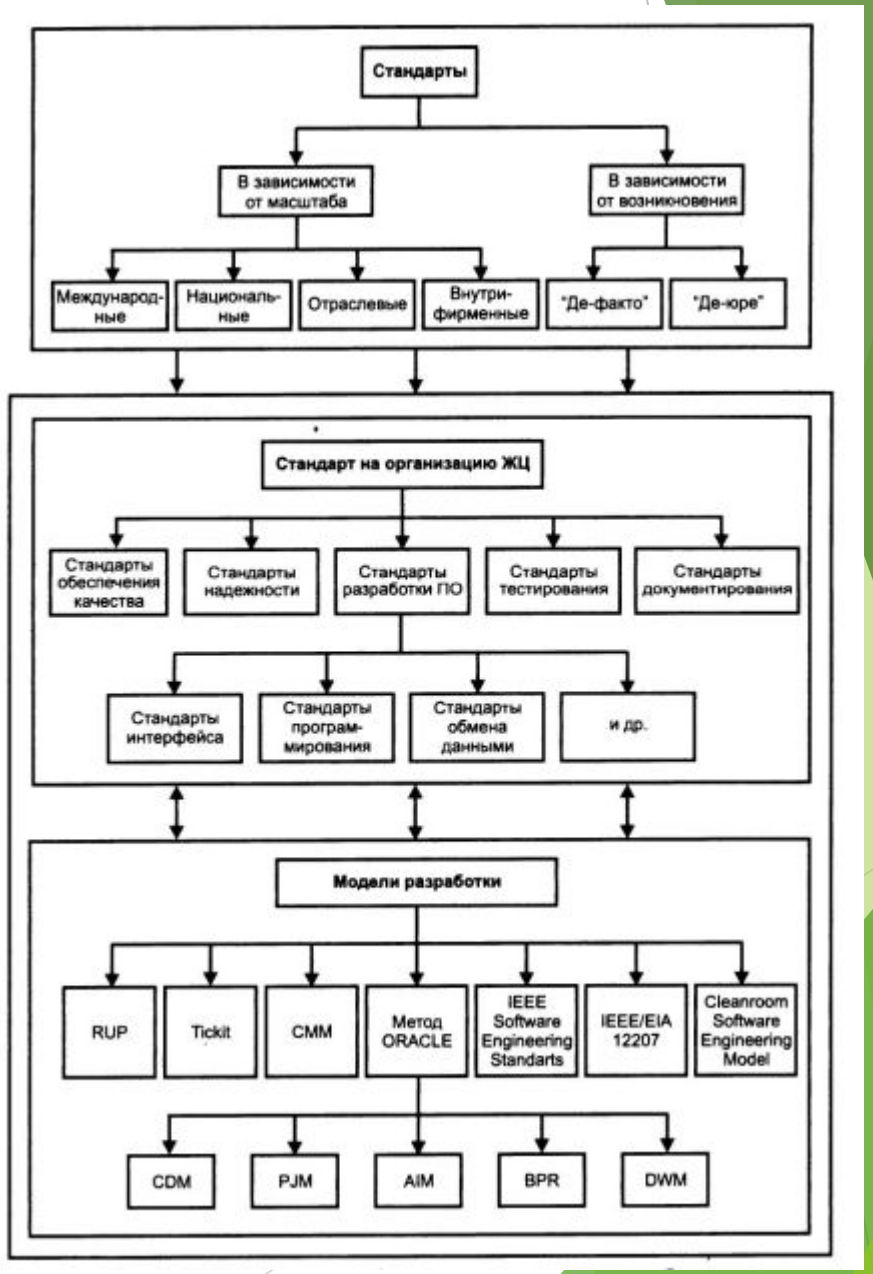
- **Стандарт** – это нормативный документ, разработанный на основе консенсуса, утвержденный признанным органом, направленный на достижение оптимальной степени упорядочения в определенной области.
- **Предварительный стандарт** – это временный документ, который принимается органом по стандартизации и доводится до широкого круга потенциальных потребителей, а так же до тех, кто может его применить.
- **Документ технических условий** – документ, который устанавливает технические требования к продукции, услуге, процессу.
- **Свод правил** – разрабатывается для процессов проектирования, монтажа оборудования и конструкций, технического обслуживания или эксплуатации объектов, конструкций, изделий.
- **Регламент** – это документ, в котором содержатся обязательные правовые нормы.

Так же нормативными документами являются:

- **ПР**- правила по стандартизации
- **Р** – рекомендации по стандартизации
- **ТУ** – технические условия.

Стандарты в области программного обеспечения

- **Стандарт «де-факто»** – продукт какого либо поставщика, который захватил большую долю рынка и который другие поставщики стремятся эмулировать, копировать или использовать для того, чтобы захватить свою часть рынка.
- **Стандарт «де-юре»** - это стандарт создающийся формально признанной стандартизирующей организацией, он разрабатывается при соблюдении правил консенсуса в процессе открытой дискуссии, в которой каждый имеет шанс принять участие.



внутрифирменных стандартов

Внутрифирменные стандарты регламентируют технологические процессы, происходящие внутри фирмы

Внутрифирменный стандарты бывают:

- Производственные – те стандарты, которые регламентируют процессы производства программного обеспечения по этапам и стадиям жизненного цикла.
- Управленческие – стандарты, регламентирующие порядок управления производственным процессом.

С помощью внутрифирменных стандартов:

- Достигаются лучшие показатели обучения персонала.
- Повышается надежность и качество программного обеспечения.
- Повышается дружественность программного продукта, сокращаются сроки обучения конечного пользователя.
- Улучшается обслуживание, сокращаются сроки внедрения программного продукта.

Жизненный цикл программных средств

- Жизненный цикл программных средств представляет собой набор этапов, частных работ и операций в последовательности их выполнения и взаимосвязи, регламентирующих ведение работ от подготовки технического задания до завершения испытаний ряда версий и окончания эксплуатации ПС или ИС.

Стандарты включают в себя :

- Правила описания исходной информации
- Способы и методы выполнения операций

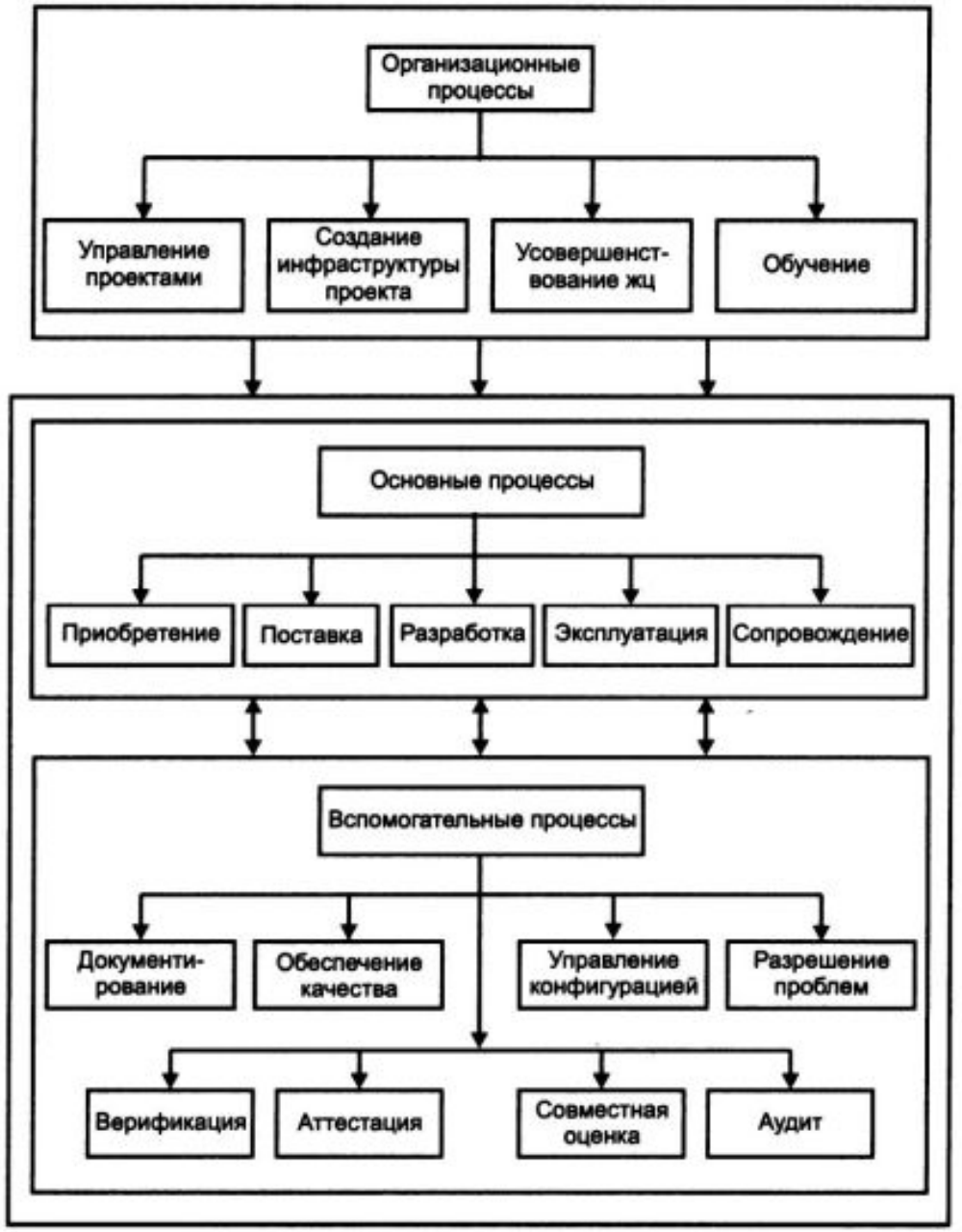
Стандарты устанавливают :

- Правила контроля технологических процессов
- Требования к оформлению результатов

Стандарты регламентируют содержание технологических и эксплуатационных документов на комплексы программ. Они определяют организационную структуру коллектива, обеспечивают распределение и планирование заданий, а так же контроль за ходом создания ПС.

- **Базовый стандарт** – принятый нормативный документ, регламентирующий типовые требования, нормы и правила применительно к данному объекту стандартизации.
- **Профиль стандарта** – принятый нормативный документ, регламентирующий требования, нормы и правила, выбранные из базовых стандартов и при необходимости дополненные и\или уточнённые применительно к конкретной классификационной группе данного объекта стандартизации.
- ГОСТ Р ИСО/МЭК 12207 (01.07.2000г.) – «процессы жизненного цикла программных средств»
- ГОСТ 34.601-90 «Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания»
- ГОСТ 34.602-89 «Информационная технология. Комплекс стандартов на автоматизированных системах. Техническое задание на создание автоматизированной системы»
- ГОСТ 34.603-92 «Информационная технология. Виды испытаний автоматизированных систем»

Процессы жизненного цикла



Основные процессы жизненного цикла программного средства

- **Процесс приобретения** состоит из действий и задач заказчика, приобретающего пс.

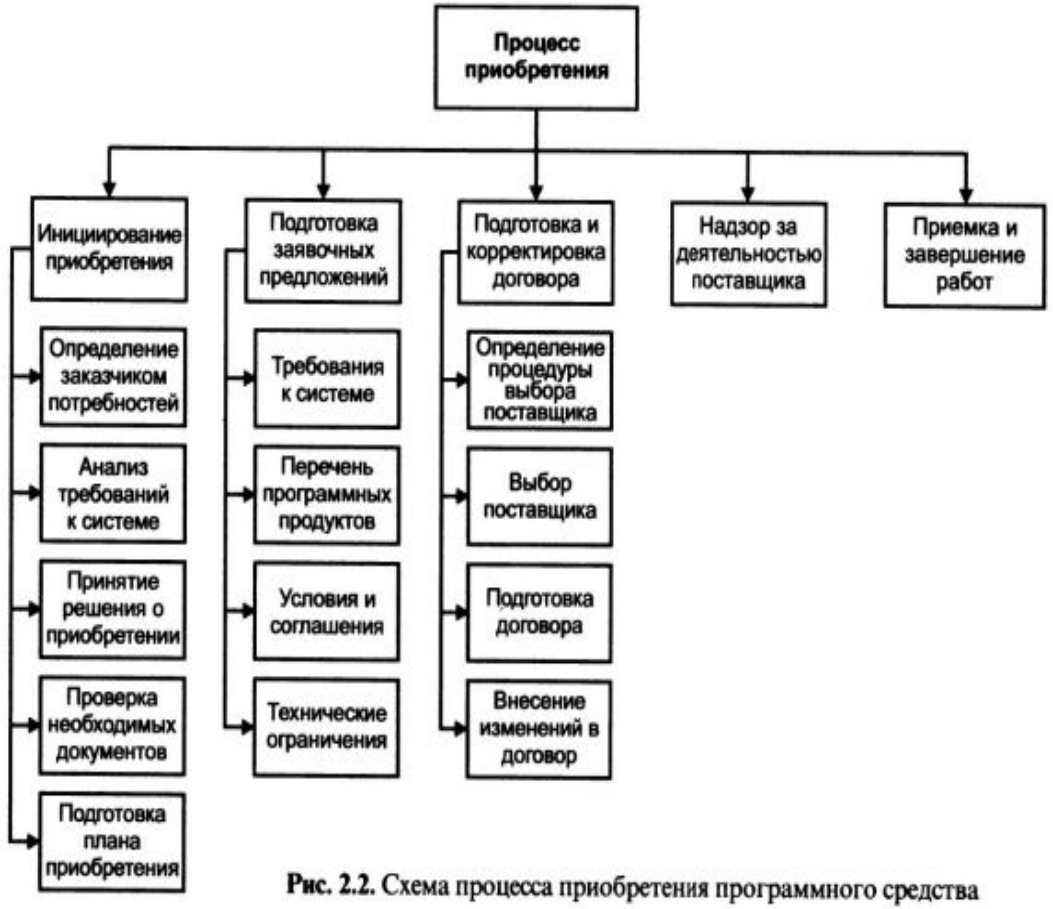


Рис. 2.2. Схема процесса приобретения программного средства

- **Процесс поставки** охватывает действия и задачи, выполняемые поставщиком, который снабжает заказчика программы продуктом или услугой.

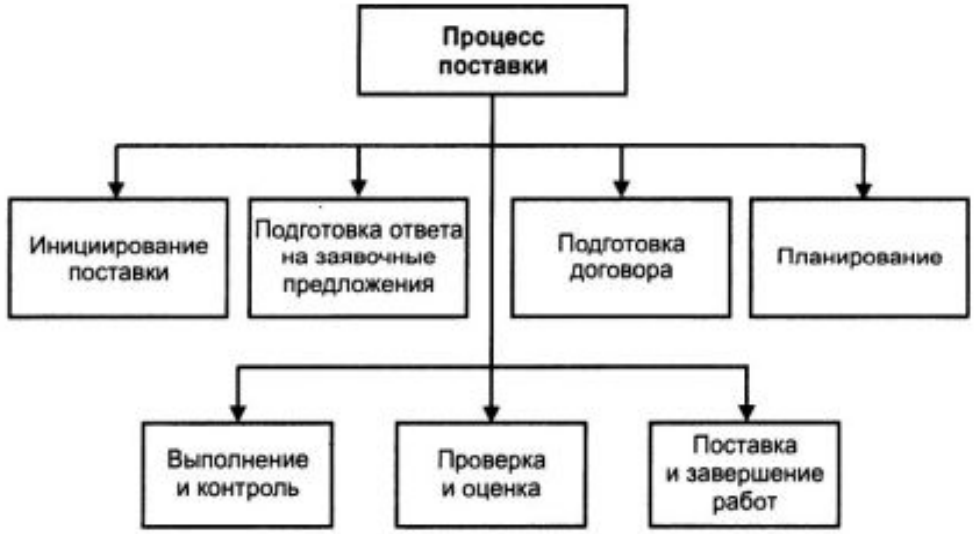


Рис. 2.3. Схема процесса поставки

- **Процесс разработки** предусматривает действия и задачи, выполняемые разработчиком, и охватывает работы по созданию ПС и его компонентов в соответствии и заданными требованиями, включая оформление проектной и эксплуатационной документации.

Рис. 2.4. Схема процесса разработки

- **Процесс эксплуатации** охватывает действия и задачи оператора - организации, эксплуатирующей систему.

Рис. 2.5. Схема процесса эксплуатации

- **Процесс сопровождения** предусматривает действия и задачи, выполняемые сопровождающей организацией.

Рис. 2.6. Схема процесса сопровождения

Вспомогательные процессы ЖЦ ПС

- **Процесс документирования** предусматривает формализованное описание информации, созданной в течение ЖЦ ПС.

Рис. 2.7. Схема процесса документирования

- **Процесс управления конфигурацией** предполагает применение административных и технических процедур на всем протяжении ЖЦ ПС

Рис. 2.8. Схема процесса управления конфигурацией

- **Процесс обеспечения качества** обеспечивает соответствующие гарантии того, что ПС и процессы его ЖЦ соответствуют заданным требованиям и утвержденным планам.

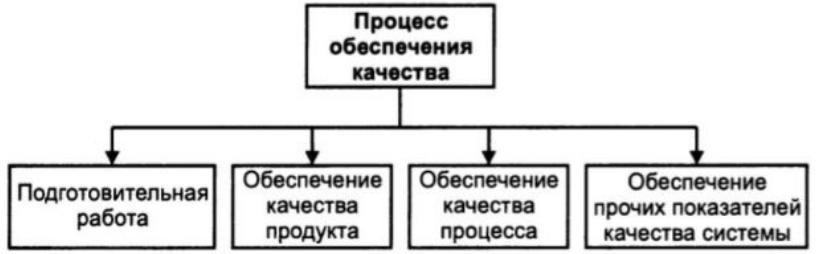


Рис. 2.9. Схема процесса обеспечения качества

- **Процесс верификации** состоит в определении того, что программные продукты, являющиеся результатами некоторого действия, полностью удовлетво овиям, обусловле действиям.

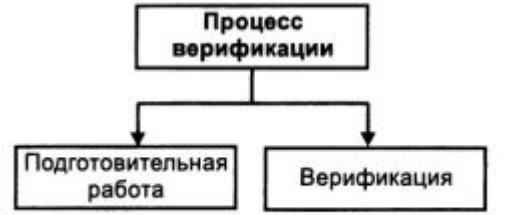


Рис. 2.10. Схема процесса верификации

- **Процесс аттестации** предусматривает определение полноты соответствия заданных требований и созданной системы или программного продукта их конкретному функциональному назначению.

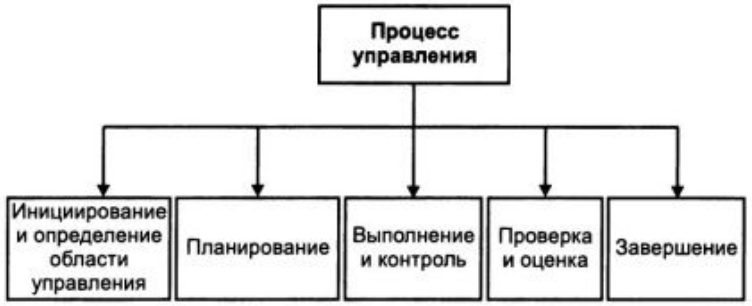
- **Проц** предназначен для оценки состояния работ по проекту и ПС, создаваемому при выполнении данных работ.

- **Процесс аудита** представляет собой определение соответствия требованиям, планам и условиям договора.

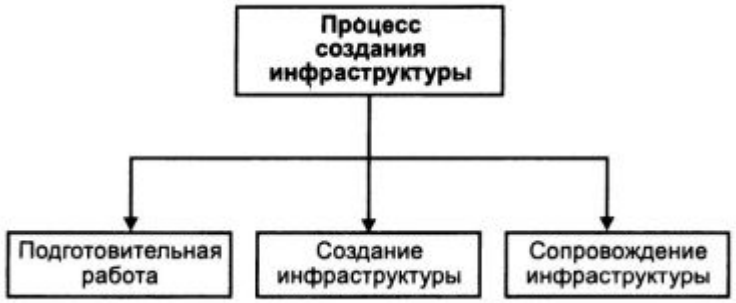
- **Процесс разрешения проблем** предусматривает анализ и решение проблем независимо от их происхождения или источника, которые обнаружены в ходе разработки

Организационные процессы ЖЦ ПС

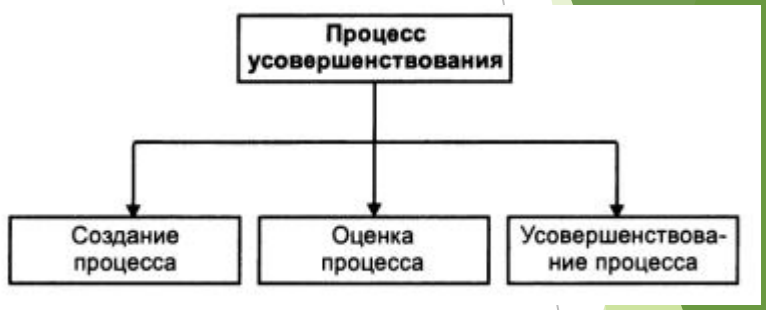
- **Процесс управления** состоит из действий и задач, которые могут выполнятся любой стороной, управляющей своими процессами.



- **Процесс создания инфраструкторы** охватывает выбор и поддержку технологии, стандартов и инструментальных средств, выбор и установку аппаратных и программных средств, используемых для разработки, эксплуатации и сопровождения ПС.



- **Процесс усовершенствования** предусматривает оценку, измерение, контроль и усовершенствование процессов ЖЦ.



- **Процесс обучения** охватывает первоначальное обучение и последующее постоянное повышении квалификации персонала.

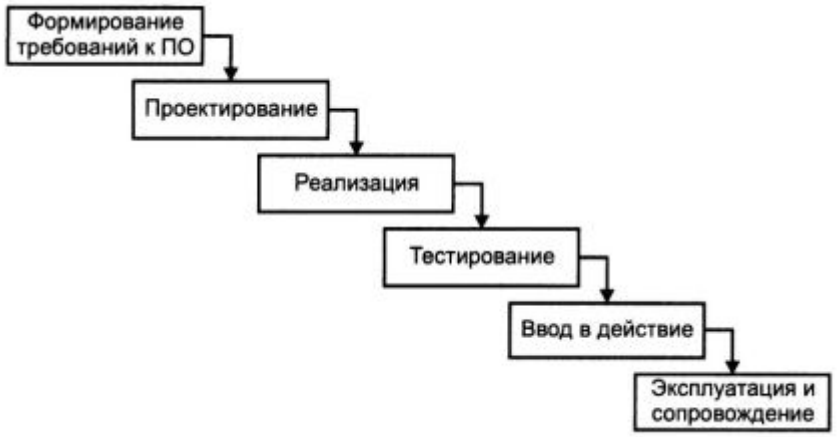
Стандарты комплекса ГОСТ 34 и Стандарт IEEE 1074-1995

- ГОСТ 34 уделяет внимание содержанию проектных документов.
- ГОСТ 34.601-90 « Стадии создания АС»
- ГОСТ 34.602-89 « ТЗ на создание АС»
- РД 50 50-34.689-90 « Требования к содержанию документов»
- IEEE 1074-1995 охватывает полный ЖЦ ПС, в котором выделяются 6 крупных базовых процессов.

Модели ЖЦ ПС

- Модель ЖЦ: структура, состоящая из процессов, робот и задач, включающих в себя разработку, эксплуатацию и сопровождение программного продукта, охватывающая жизнь системы от установления требований к ней до прекращения ее использования.

Каскадный способ: разбиение всей разработки на этапы, причем переход с одного этапа на следующий происходит только после того, как будет полностью завершена работа на текущем



Спиральная модель: делает упор на начальные этапы ЖЦ (анализ и проектирование)
Основная проблема спирального метода-определение момента перехода на следующий этап.

Единая система программной документации

- ЕСПД – это комплекс государственных стандартов, устанавливающих взаимоувязанные правила разработки, оформления и обращения программ и программной документации.
- Стандарты ЕСПД подразделяют на группы :

Код группы	Наименование группы
0	Общие положения
1	Основополагающие стандарты
2	Правила выполнения документации разработки
3	Правила выполнения документации изготовления
4	Правила выполнения документации сопровождения
5	Правила выполнения эксплуатационной документации
6	Правила обращения программной документации
7	Резервные группы
8	
9	Прочие стандарты

ГОСТ 19.101-77 ЕСПД « Виды программ и программных документов»

Гост подразделяет программ на следующие виды:

- Компонент
- Комплекс

Виды программных документов и их содержание:

- Спецификация
- Ведомость держателей подлинников
- Текст программы
- Описание программы
- Программа и методика испытаний
- Техническое задание
- Пояснительная записка
- Эксплуатационные документы

ГОСТ 19.102-77 ЕСПД « Стадии разработки»

Стадии разработки и этапы работы:

- (разработка и утверждение технического задания)
- Эскизный проект(разработка эскизного проекта, утверждение эскизного проекта)
- Технический проект (разработка технического проекта, утверждение технического проекта)
- Рабочий проект (разработка программы, разработка программной документации, испытание программы)
- Внедрение (подготовка и передача программы)

ГОСТ 19.105-78 ЕСПД « Общие требования к программным документам»

Программный документ состоит из следующих условных частей:

- Титульной(оформляется согласно ГОСТ 19.104-78)
- Информационной (должна состоять из аннотации и содержания)
- Основной

ГОСТ 19.201-78 ЕСПД. «Техническое задание, Требования к содержанию и оформлению»

Техническое задание должно содержать следующие разделы:

- Введение
- Основания для разработки
- Назначение разработки
- Требования к программе или программному изделию
- Требования к программной документации
- Технико-экономические показатели
- Стадии и этапы разработки
- Порядок контроля и приемки
- Допускается включать приложения

Требования к программной документации:

- Предварительный состав программной документации
- Специальные требования к программной документации

ГОСТ 19.402-78 « Описание программы»

Данный стандарт определяет состав и требования к содержанию программного документа «Описание программы»

Описание программы включает:

- Общие сведения
- Функциональное назначение
- Описание логической структуры
- Используемые технические средства
- Вызов и загрузка
- Входные данные
- Выходные данные

ГОСТ 19.404-79 ЕСПД «Пояснительная записка. Требования к содержанию и оформлению»

Пояснительная записка должна включать следующие разделы:

- Введение
- Назначение и область применения
- Технические характеристики
- Ожидаемые технико-экономические показатели
- Источники, использованные при разработке

ГОСТ 19.503-79 «Руководство системного программиста. Требования к содержанию и оформлению»

Руководство системного программиста должно содержать следующие разделы:

- Общие сведения о программе
- Структура программы
- Настройка программы
- Проверка программы
- Дополнительные возможности
- Сообщения системному программисту

ГОСТ 19.504-79 ЕСПД « Руководство программиста. Требования к содержанию и оформлению»

Руководство программиста должно содержать разделы:

- Назначение и условия применения программы
- Характеристики программы
- Обращения к программе
- Входные и выходные данные
- Сообщения

ГОСТ 19.505-79 «Руководство оператора. Требования к содержанию и оформлению»

Руководство оператора должно включать:

- Назначение программы
- Условия выполнения программы
- Выполнение программы
- Сообщения оператору

ГОСТ 19.506-79 ЕСПД «Описание языка. Требования к содержанию и оформлению»

При описании языка необходимо узнать:

- Общие сведения
- Элементы языка

Кроме того, допускается вводить дополнительные разделы.

- Способы структурирования программы
- Средства обмена данными
- Встроенные элементы
- Средства отладки программы

Перечень документов ЕСПД

- ГОСТ 19.001-77 ЕСПД « Общие положения»
- ГОСТ 19.005-85 ЕСПД « Р-схемы алгоритмов и программ. Обозначения условные графические и правила управления.
- ГОСТ 19.101-77 ЕСПД « Виды программ и программных документов»
- ГОСТ 19.102-77 ЕСПД « Стадии разработки»
- ГОСТ 19.103-77 ЕСПД « Обозначение программ и программных документов»
- ГОСТ 19.104-78 ЕСПД « Основные надписи»
- ГОСТ 19.105-78 ЕСПД «Общие требования к программным документам»
- ГОСТ 19.106-78 ЕСПД «Требования к программным документам, выполненным печатным способом»
- ГОСТ 19.201-78 ЕСПД «Технические задания. Требования к содержанию и оформлению»
- ГОСТ 19.202-78 ЕСПД « Спецификация. Требования к содержанию и оформлению»
- ГОСТ 19.301-79 ЕСПД «Порядок и методика испытаний»
- ГОСТ 19.401-78 ЕСПД « Текст программы. Требования к содержанию и оформлению»
- ГОСТ 19.402-78 ЕСПД « Описание программ»
- ГОСТ 19.403-79 ЕСПД « Ведомость держателей подленников»

- ГОСТ 19.404-79 ЕСПД « Пояснительная записка. Требования к содержанию и оформлению»
- ГОСТ 19.501-78 ЕСПД «Формуляр. Требования к содержанию и оформлению»
- ГОСТ 19.502-78 ЕСПД « Описание применения. Требования к содержанию и оформлению»
- ГОСТ 19.503-79 ЕСПД « Руководство системного программиста. Требования к содержанию и оформлению»
- ГОСТ 19.504-79 ЕСПД « Руководство программиста. Требования к содержанию и оформлению»
- ГОСТ 19.505-79 ЕСПД « Руководство оператора. Требования к содержанию и оформлению»
- ГОСТ 19.506-79 ЕСПД « Описание языка. Требования к содержанию и оформлению»
- ГОСТ 19.507-79 ЕСПД « Ведомость эксплуатационных документов»
- ГОСТ 19.508-79 ЕСПД « Руководство по техническому обслуживанию. Требования к содержанию и оформлению»
- ГОСТ 19.601-78 ЕСПД « Общие правила дублирования, учета и хранения»
- ГОСТ 19.602-78 ЕСПД « Правила дублирования, учета и хранения программных документов, выполненных печатным образом»
- ГОСТ 19.603-78 ЕСПД « Общие правила внесения изменения»
- ГОСТ 19.604-78 ЕСПД « Правила внесения изменений в программные документы, выполняемые печатным способом»
- ГОСТ 19.701-90 ЕСПД « Схема алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения»
- ГОСТ 19781-90 « Программное обеспечение систем обработки информации. Термины и определения»

Государственные стандарты рф (гост р)

- ГОСТ Р ИСО/МЭК 9294-93. Информационная технология. Руководство по управлению документированием программного обеспечения.
- ГОСТ Р ИСО/МЭК 9126-93. Информационная технология. Оценка программной продукции. Характеристики качества и руководства по их применению.
- ГОСТ Р ИСО 9127. Системы обработки информации. Документация пользователя и информация на упаковке для потребительских программных пакетов.
- ГОСТ Р ИСО/МЭК 8631-94. Информационная технология. Программные конструктивы и условные обозначения для их представления.
- ГОСТ Р ИСО/МЭК 12119:1994. Информационная технология. Пакеты программных средств. Требования к качеству и испытания.

Документация пользователя

Документация пользователя – полный комплект документов, поставляемых в печатном или другом виде, который обеспечивает применение продукта, а так же является его неотъемлемой частью.

Документация пользователя должна отвечать следующим характеристикам:

- Полнота
- Правильность
- Непротиворечивость
- Понятность
- Простота обозрения

Программы и данные

Функциональные возможности:

- Установка
- Реализация функций
- Правильность
- Непротиворечивость

Надежность и качество программных средств

Для обеспечения надежности ПС в конкретных проектах должны быть организованны и стимулированы разработка, освоение и применение своевременных автоматизированных технологий и инструментальных средств, обеспечивающих:

- Предупреждение исключение дефектов и ошибок при создании ПС
- Контроль надежности и безопасности
- Стандартизация технологических процессов и объектов

Поддержка этапов и работ ЖЦ ПС

- В результате на начальном этапе проектирования следует формировать весь комплект документов-***профиль***, обеспечивающий регламентирование всех этапов и работ при создании надежных ПС
- ПС должны быть поддержаны методами и средствами систематического, автоматизированного ***тестирования и испытаний***.
- ПС должна сопутствовать ***обязательная сертификация***
- При проявлении дефектов ПС в процессе эксплуатации приводит к необходимости создания специальных ***систем автоматической оперативной защиты***

Экспериментальное определение реальной надежности функционирования сложных комплексов программ-весьма трудоёмкая, трудно автоматизируемая и не всегда безопасная часть жизненного цикла ПС.

Основные понятия и показатели надежности программных средств

Надежность-свойство объекта выполнять заданные функции, сохраняя во времени значение установленных эксплуатационных показателей в заданных пределах, соответствующих заданным режимам и условиям использования, технического обслуживания, ремонта, хранения и транспортирования.

Надежность является составной частью более общего понятия-качества.

Свойства изучаются теорией надежности.

Факторы надежности:

- Надежность компонентов
- Дефект в контракции

Надежности программных средств

К задачам теории и анализа надежности сложных программных средств можно отнести следующие:

- формулирование основных понятий;
- выявление и исследование основных факторов;
- •выбор и обоснование критериев надежности для комплексов программ различного типа и назначения;
- исследование дефектов и ошибок;
- исследование и разработка методов структурного построения сложных ПС;
- исследование методов и средств контроля и защиты от искажений программ;
- разработка методов и средств определения и прогнозирования характеристик надежности в жизненном цикле комплексов программ с учетом их функционального назначения, сложности, структурного построения и технологии разработки.

Методы и средства контроля и диагностики функционирования системы

Методы и средства диагностического контроля предназначены для установления степени работоспособности системы, локализации отказов, определения их характеристик и причин, скорейшего восстановления работоспособности, для накопления, обобщения и анализа данных, характеризующих работоспособность системы.

Основные задачи технической диагностики включают *в* себя:

- контроль исправности системы и полного соответствия ее состояния и функций технической документации;
- проверку работоспособности системы и возможности выполнения всех функций в заданном режиме работы в любой момент времени с характеристиками, заданными технической документацией;
- поиск, выявление и локализацию источников и результатов сбоев, отказов и неисправностей в системе.

Показатели качества и надежности программных средств.

В международном стандарте ISO 9126:1991 при отборе минимума стандартизируемых показателей выдвигались и учитывались следующие принципы: я сность и измеряемость значений, отсутствие перекрытия между используемыми показателями, соответствие установившимся понятиям и терминологии, возможность последующего уточнения и детализации.

Функциональная пригодность детализируется пригодностью для применения, точностью, защищенностью, способностью к взаимодействию и согласованностью со стандартами и правилами проектирования.

6 основных характеристик качества ПС

Надежность рекомендуется характеризовать уровнем завершенности (отсутствия ошибок), устойчивостью к ошибкам и перезапускаемостью.

Применимость предлагается описывать понятностью, обучаемостью и простотой использования.

Эффективность рекомендуется характеризовать ресурсной и временной экономичностью.

Сопровождаемостъ характеризуется удобством для анализа, изменяемостью, стабильностью и тестируемостью.

Переносимость предлагается отражать адаптируемостью, структурированностью, замещаемостью и внедряемостью.

ГОСТ 28195-89

В стандарте ***ГОСТ 28806-90*** формализуются общие понятия программы, программного средства, программного продукта и их качества.

Функциональная пригодность — это набор атрибутов, определяющий назначение, номенклатуру, основные необходимые и достаточные функции ПС, заданные техническим заданием заказчика или потенциального пользователя.

В наиболее общем виде функциональная пригодность проявляется в ***корректности и надежности*** **ПС.**

Понятие ***корректной (правильной) программы*** может рассматриваться статически вне ее исполнения во времени.

Надежная программа, прежде всего, должна обеспечивать достаточно низкую вероятность отказа в процессе функционирования в реальном времени.

Принцип классификации сбоев и отказов

При длительности восстановления, меньшей заданного порога, дефекты и аномалии при функционировании программ следует относить к ***сбоям,*** а при восстановлении, превышающем по длительности пороговое значение, происходящее искажение соответствует ***отказу.***

Надежность функционирования ПС наиболее широко характеризуется ***устойчивостью,*** или способностью к безотказному функционированию, и ***восстанавливаемостью*** работоспособного состояния после произошедших сбоев или отказов.

Восстанавливаемость характеризуется полнотой и длительностью восстановления функционирования программ в процессе перезапуска — рестарта.

Дестабилизирующие факторы и методы обеспечения надежности функционирования программных средств

При любом виде деятельности людям свойственно непредумышленно ошибаться, результаты чего проявляются в процессе создания или применения изделий или систем. В общем случае под ошибкой подразумевается дефект, погрешность или неумышленное искажение объекта или процесса. При этом предполагается, что известно правильное, эталонное состояние объекта, по отношению к которому может быть определено наличие отклонения — ***дефекта или ошибки.***

Объекты уязвимости

- динамический вычислительный процесс обработки данных, автоматизированной подготовки решений и выработки управляющих воздействий;
- информация, накопленная в базах данных, отражающая объекты внешней среды, и процессы ее обработки;
- объектный код программ, исполняемых вычислительными средствами в процессе функционирования ПС;
- информация, выдаваемая потребителям и на исполнительные механизмы, являющаяся результатом обработки исходных данных и информации, накопленной в базе данных.

На эти объекты воздействуют различные ***дестабилизирующие факторы,*** которые можно разделить на внутренние, присущие самим объектам уязвимости, и внешние, обусловленные средой, в которой эти объекты функционируют.

Внутренними источниками угроз надежности функционирования сложных ПС можно считать следующие дефекты программ:

- системные ошибки при постановке целей и задач создания ПС, при формулировке требований к функциям и характеристикам решения задач, определении условий и параметров внешней среды, в которой предстоит применять ПС;
- алгоритмические ошибки разработки при непосредственной спецификации функций программных средств, при определении структуры и взаимодействия компонентов комплексов программ, а также при использовании информации баз данных:
- ошибки программирования в текстах программ и описаниях данных, а также в исходной и результирующей документации на компоненты и ПС в целом;
- недостаточную эффективность используемых методов и средств оперативной защиты программ и данных от сбоев и отказов и обеспечения надежности функционирования ПС в условиях случайных негативных воздействий.

Внешними дестабилизирующими факторами, отражающимися на надежности функционирования перечисленных объектов уязвимости в ПС, являются:

- ошибки оперативного и обслуживающего персонала в процессе эксплуатации ПС;
- искажения в каналах телекоммуникации информации, поступающей от внешних источников и передаваемой потребителям, а также недопустимые для конкретной информационной системы характеристики потоков внешней информации:
- **сбои и** отказы в аппаратуре вычислительных средств;
- изменения состава и конфигурации комплекса взаимодействующей аппаратуры информационной системы за пределы, проверенные при испытаниях или сертификации и отраженные в эксплуатационной документации.

Методы обеспечения надежности программных средств.

- ***создавать программные модули и функциональные компоненты*** высокого, гарантированного качества;
- ***предотвращать дефекты проектирования*** за счет эффективных технологий и средств автоматизации обеспечения всего жизненного цикла комплексов программ и баз данных;
- ***обнаруживать и устранять различные дефекты и ошибки*** проектирования, разработки и сопровождения программ путем систематического тестирования на всех этапах жизненного цикла ПС;
- ***удостоверять достигнутое качество и надежность функционирования*** **ПС** в процессе их испытаний и сертификации перед передачей в регулярную эксплуатацию;
- ***оперативно выявлять последствия дефектов программ и данных*** и восстанавливать нормальное, надежное функционирование комплексов программ.

Все принципы и методы обеспечения надежности в соответствии с их целью можно разбить на четыре группы:

- предупреждение ошибок
- обнаружение ошибок
- исправление ошибок
- обеспечение устойчивости к ошибкам.

К первой группе относятся принципы и методы, позволяющие минимизировать или вообще исключить ошибки. Методы второй группы сосредоточивают внимание на функциях самого программного обеспечения, помогающих выявлять ошибки. К третьей группе относятся функции программного обеспечения, предназначенные для исправления ошибок или их последствий. Устойчивость к ошибкам (четвертая группа) — это мера способности системы программного обеспечения продолжать функционирование при наличии ошибок.

Модели надежности программного обеспечения

Термин ***модель надежности программного обеспечения***, как правило, относится к математической модели, построенной для оценки зависимости надежности программного обеспечения от некоторых определенных параметров.

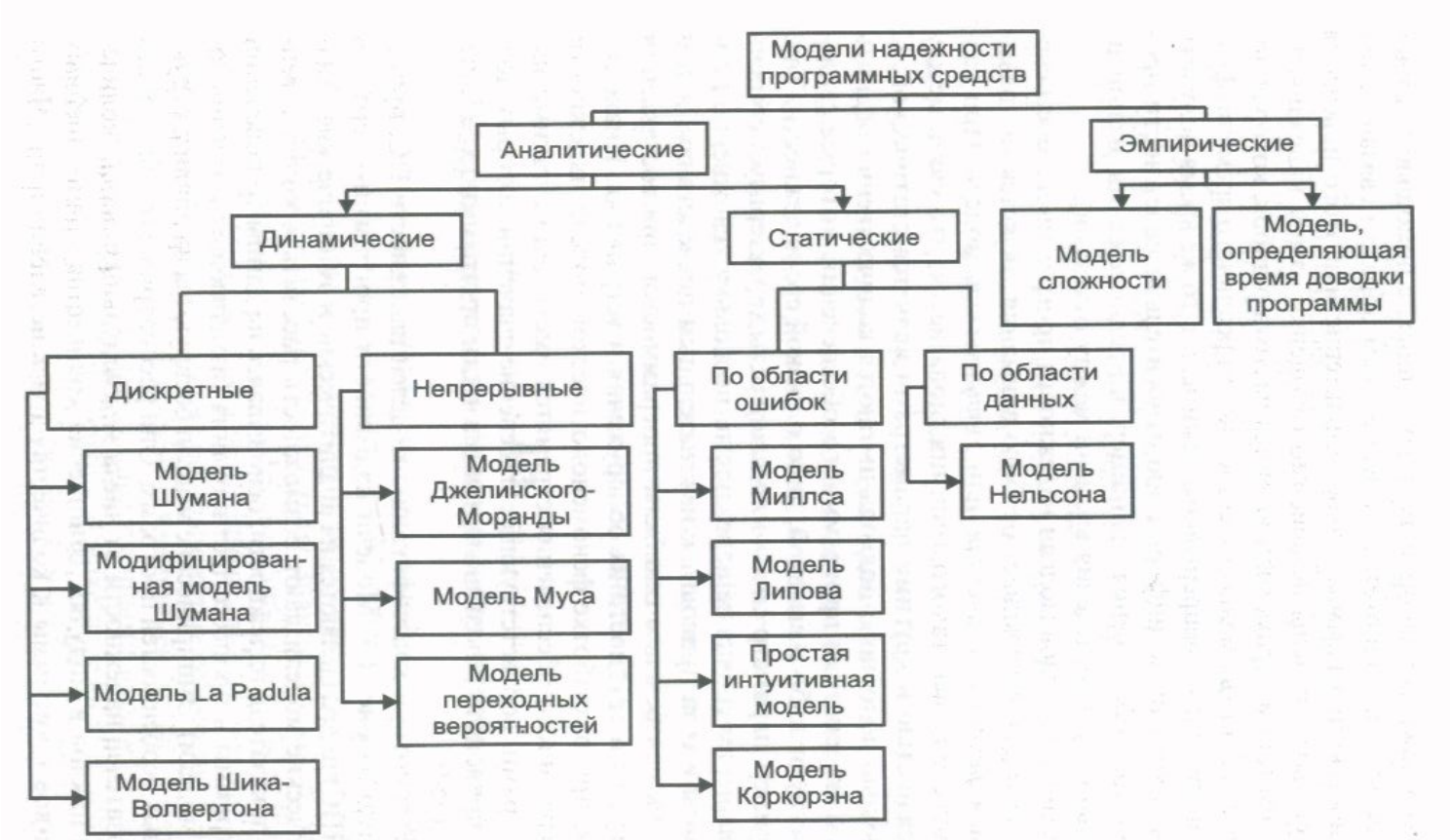


Рис. 4.3. Классификационная схема моделей надежности ПС

Обеспечение качества и надежности в процессе разработки сложных программных средств

- ***Сложность***
- **Отношения с пользователем**
- **Решение задачи**
- **Поймите задачу**
- **Составьте план**
- **Выполните план**
- **Проанализируйте решение**

Требования к технологии и средствам автоматизации разработки сложных программных средств

Для обеспечения качества и надежности ПС стандартами ***рекомендуется формулировать требования:***

- к объекту разработки на данном этапе — к его программным и информационным компонентам, а также к интерфейсу между ними и внешней средой;
- к процессу, технологии и организации выполнения совокупности работ и документов каждого этапа;
- к методам и характеристикам средств автоматизации выполнения работ, обеспечивающим необходимую надежность функционирования и качество ПС;
- к методам и средствам контроля, измерения и документирования качества процессов и результатов выполненных работ.

Требования к инструментальным средствам автоматизации разработки надежных ПС наиболее полно изложены в стандарте ***IEEE 1209-1992.***

Качество программного обеспечения

Качество — совокупность характеристик объекта, относящихся к его способности удовлетворить установленные и предполагаемые потребности.

Можно выделить три большие группы факторов, влияющих на качество программного обеспечения:

- *функциональная* — связана с полнотой и удобством использования реализованных функций программного средства;
- *административная* — связана с квалификацией персонала, организационной структурой и управлением персоналом;
- *программно-архитектурная* — связана с процессом разработки программного обеспечения, выбранными методологиями, инструментальными средствами, использованными на различных этапах жизненного цикла программного обеспечения, а также архитектурой программного средства.

ТЕСТИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА

Основные определения:

Тестирование (testing) — процесс выполнения программы (или части программы) с намерением (или целью) найти ошибки.

Доказательство (proof) — попытка найти ошибки в программе безотносительно к внешней для программы среде.

Контроль (verification) — попытка найти ошибки, выполняя программу в тестовой, или моделируемой, среде.

Испытание (validation) — попытка найти ошибки, выполняя программу в заданной реальной среде.

Аттестация (certification) — авторитетное подтверждение правильности программы. При тестировании с целью аттестации выполняется сравнение с некоторым заранее определенным стандартом.

Отладка (debugging) не является разновидностью тестирования. Хотя слова «отладка» и «тестирование» часто используются как синонимы, под ними подразумеваются разные виды деятельности.

Тестирование модуля, или автономное тестирование *(module testing, unit testing),* — контроль отдельного программного модуля, обычно в изолированной среде (т. е. изолированно от всех остальных модулей). Тестирование модуля иногда включает также математическое доказательство.

Тестирование сопряжений *(integration testing)* — контроль сопряжений между частями системы (модулями, компонентами, подсистемами).

Тестирование внешних функций (external function testing) — контроль внешнего поведения системы, определенного внешними спецификациями.

Комплексное тестирование *(system testing)* — контроль и/или испытание системы по отношению к исходным целям. Комплексное тестирование является процессом контроля, если оно выполняется в моделируемой среде, и процессом испытания, если выполняется в среде реальной, жизненной.

Тестирование приемлемости *(acceptance testing)* — проверка соответствия программы требованиям пользователя.

Тестирование настройки *(installation testing)* — проверка соответствия каждого конкретного варианта установки системы с целью выявить любые ошибки, возникшие в процессе настройки системы.

Тестирование программы как «черного ящика»

Одним из способов изучения поставленного вопроса является исследование стратегии тестирования, называемой стратегией «черного ящика», *тестированием с управлением по данным* или *тестированием с управлением по входу-выходу.*

При таком подходе обнаружение всех ошибок в программе является критерием *исчерпывающего входного тестирования.*программы не соответствует спецификации.

Построение исчерпывающего входного теста невозможно. Это подтверждается двумя аргументами: во-первых, нельзя создать тест, гарантирующий отсутствие ошибок; во-вторых, разработка таких тестов противоречит экономическим требованиям.

Тестирование программы как «белого ящика»

Стратегия *«белого ящика»,* или стратегия тестирования, *управляемого логикой программы,* позволяет исследовать внутреннюю структуру программы. В этом случае тестирующий получает тестовые данные путем анализа логики программы (к сожалению, здесь часто не используется спецификация программы).

Исчерпывающему входному тестированию может быть поставлено в соответствие *исчерпывающее тестирование маршрутов.*

Аксиомы (принципы) тестирования

- *Хорош тот тест, для которого высока вероятность обнаружить ошибку*
- *Одна из самых сложных проблем при тестировании — решить, когда нужно его закончить.*
- *Не нужно тестировать свою собственную программу.*
- *Необходимая часть всякого теста — описание ожидаемых выходных данных или результатов.*
- *Избегайте невоспроизводимых тестов, не тестируйте «с лету».*
- *Готовьте тесты как для правильных, так и для неправильных входных данных.*

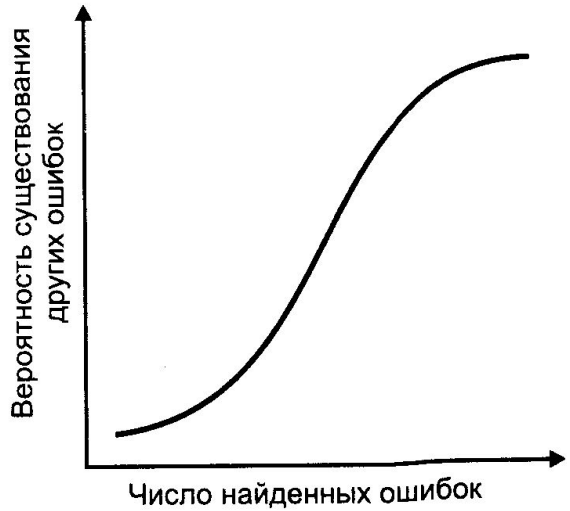


Рис. 5.1. График соотношения между обнаруженными и необнаруженными ошибками

- *Детально изучите результаты каждого теста.*
- *По мере того как число ошибок, обнаруженных в некотором компоненте программного обеспечения, увеличивается, растет относительная вероятность существования в нем необнаруженных ошибок.*
- *Поручайте тестирование самым способным программистам.*
- *Проект системы должен быть таким, чтобы каждый модуль подключался к системе только один раз.*
- *Никогда не изменяйте программу, чтобы облегчить ее тестирование.*
- *Тестирование, как почти всякая другая деятельность, должно начинаться с постановки целей.*

Приведем еще раз три наиболее важных принципа тестирования.

1. *Тестирование — это процесс выполнения программ с целью обнаружения ошибок.*
2. *Хорошим считается тест, который имеет высокую вероятность обнаружения еще не выявленной ошибки.*
3. *Удачным считается тест, который обнаруживает еще не выявленную ошибку.*

Философия тестирования

Задачи для теста, проектирование, написание тестов, тестирование тестов, выполнение тестов и изучение результатов тестирования. Решающую роль играет проектирование теста.

Решающую роль играет проектирование теста. Возможен целый спектр подходов к выработке философии, или стратегии проектирования. Чтобы ориентироваться в стратегиях проектирования тестов, стоит рассмотреть два крайних подхода, находящихся на границах спектра. Следует отметить также, что многие из тех, кто работает в этой области, часто бросаются в одну или другую крайность.

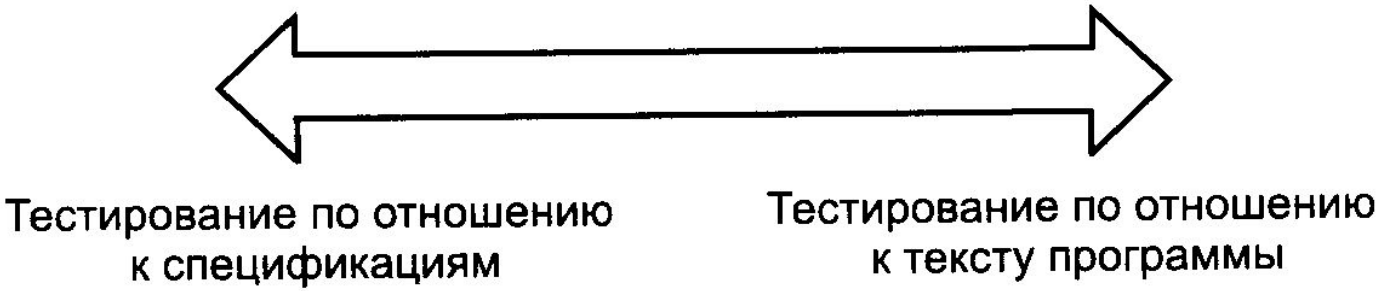
программа рассматривается как «черный ящик»

Сторонник подхода, соответствующего левой границе спектра

Программу он рассматривает как «черный ящик». Позиция его такова: «Меня не интересует, как выглядит эта программа и выполнил ли я все команды или все пути.

Я буду удовлетворен, если программа будет вести себя так, как указано в спецификациях». Его идеал — проверить все возможные комбинации и значения на входе.

тестирование — проблема в значительной степени экономическая



Тестирование модулей

Тестирование модулей (или блоков) представляет собой процесс тестирования отдельных подпрограмм или процедур программы. Здесь подразумевается, что, прежде чем начинать тестирование программы в целом, следует протестировать отдельные небольшие модули, образующие эту программу. Такой подход мотивируется тремя причинами. Во-первых, появляется возможность управлять комбинаторикой тестирования, поскольку первоначально внимание концентрируется на небольших модулях программы. Во-вторых, облегчается задача отладки программы, т.е. обнаружение места ошибки и исправление текста программы. В-третьих, допускается параллелизм, что позволяет одновременно тестировать несколько модулей.

Тестирование модулей в основном ориентировано на принцип «белого ящика».

Пошаговое тестирование

Реализация процесса тестирования модулей опирается на два ключевых положения: построение эффективного набора тестов и выбор способа, посредством которого модули комбинируются при построении из них рабочей программы.

Второе положение является важным, так как оно задает форму написания тестов модуля, типы средств, используемых при тестировании, порядок кодирования и тестирования модулей, стоимость генерации тестов и стоимость отладки.

Рассмотрим два подхода к комбинированию модулей: пошаговое и монолитное тестирование.

Детального разбора обоих методов мы делать не будем, приведем лишь некоторые общие выводы

1. Монолитное тестирование требует больших затрат труда. При пошаговом же тестировании «снизу-вверх» затраты труда сокращаются.
2. Расход машинного времени при монолитном тестировании меньше.
3. Использование монолитного метода предоставляет большие возможности для параллельной организации работы на начальной фазе тестирования (тестирования всех модулей одновременнофазе.
4. При пошаговом тестировании раньше обнаруживаются ошибки в интерфейсах между модулями, поскольку раньше начинается сборка программы.
5. Отладка программ при пошаговом тестировании легче.
6. Результаты пошагового тестирования более совершенны.

В заключение отметим, что п. 1, 4, 5, 6 демонстрируют преимущества пошагового тестирования, а п. 2 и 3 — его недостатки.

Восходящее тестирование

- При восходящем подходе программа собирается и тестируется «снизу вверх». Только модули самого нижнего уровня («терминальные» модули; модули, не вызывающие других модулей) тестируются изолированно, автономно.
- При восходящем тестировании для каждого модуля необходим *драйвер*

Нисходящее тестирование

Нисходящее тестирование (называемое также нисходящей разработкой) не является полной противоположностью восходящему,- но в первом приближении может рассматриваться как таковое. При нисходящем подходе программа собирается я тестируется «сверху вниз». Изолированно тестируется только головной модуль.

Метод «большого скачка»

- Вероятно, самый распространенный подход к интеграции модулей — метод
- Метод «большого скачка» по сравнению с другими подходами имеет много недостатков и мало достоинств. Заглушки и драйверы необходимы для каждого модуля. Модули не интегрируются до самого последнего момента, а это означает, что в течение долгого времени серьезные ошибки в сопряжениях могут остаться необнаруженными.

Метод сандвича

Тестирование методом сандвича представляет собой компромисс между восходящим и нисходящим подходами. Здесь делается попытка воспользоваться достоинствами обоих методов, избежав их недостатков.

Метод сандвича сохраняет такое достоинство нисходящего и восходящего подходов, как начало интеграции системы на самом раннем этапе.

Модифицированный метод сандвича

При тестировании методом сандвича возникает та же проблема, что и при нисходящем подходе, хотя здесь она стоит не так остро. Проблема эта в том, что невозможно досконально тестировать отдельные модули.

Комплексное тестирование

Если вы не сформулировали цели вашего продукта или если эти цели неизмеримы, вы не можете выполнить комплексное тестирование.

Комплексное тестирование может быть процессом и контроля, и испытаний. Процессом испытаний оно является тогда, когда выполняется в реальной среде пользователя или в обстановке, которая специально создана так, чтобы напоминать среду пользователя.

Проектирование комплексного теста

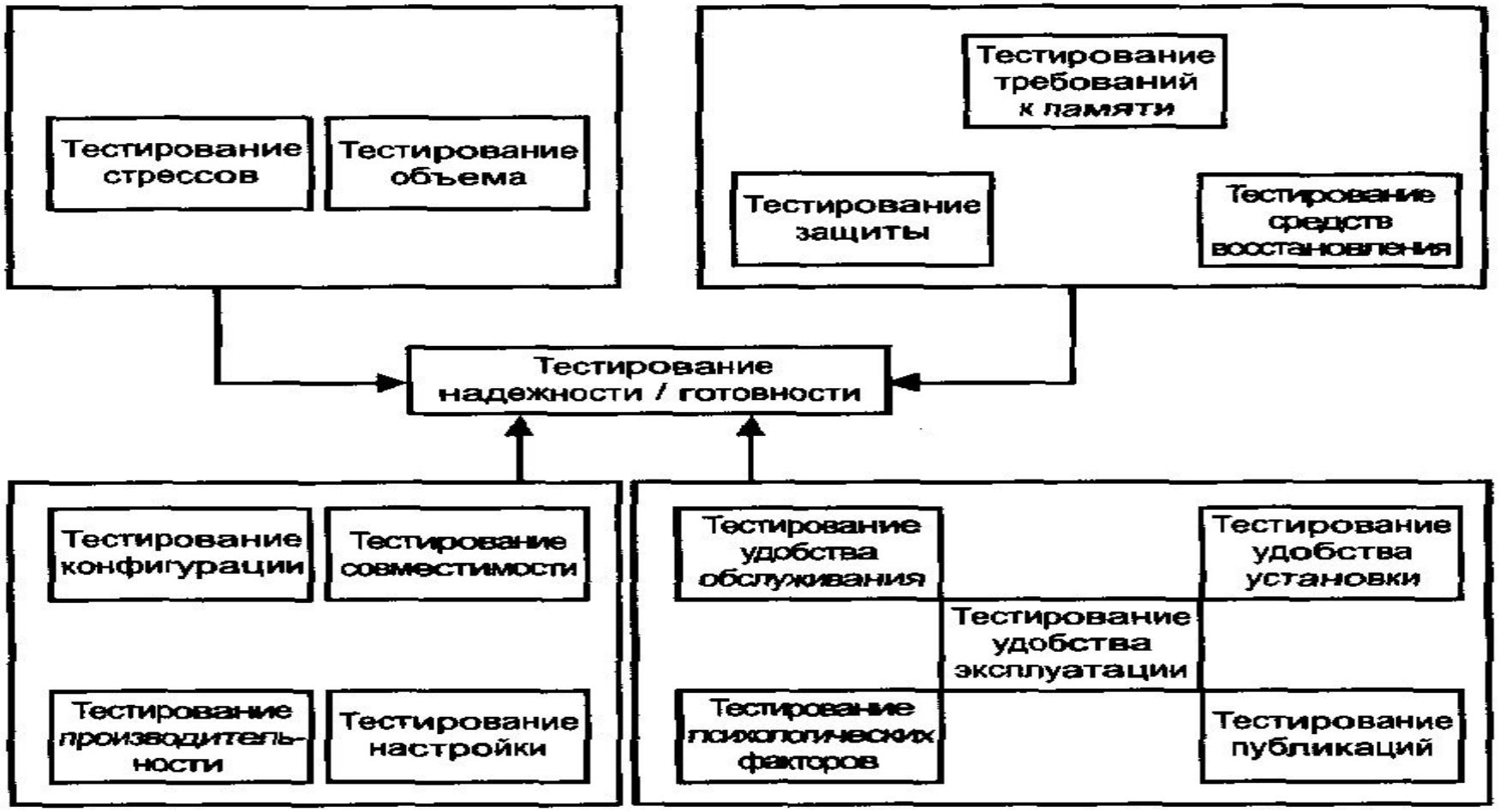


Рис. 5.3. Схема проектирования комплексного теста

Выполнение комплексного теста

- Один из методов, позволяющих вовлечь в тестирование пользователей, — *опытная эксплуатация.*
- Второй полезный метод — использовать систему в организации-изготовителе для внутренних нужд.
- При комплексном тестировании часто начинают с простых тестов, приберегая более сложные тесты к концу.

ГОСТ Р ИСО/МЭК 12119-2000

- Рассмотрим требования, которые предъявляет ***ГОСТ Р ИСО/МЭК 12119-2000*** к тестированию пакетов программ.
- ***ГОСТ Р ИСО/МЭК 12119-2000*** содержит указания, которые определяют порядок тестирования продукта на соответствие его требованиям к качеству. Эти указания охватывают как тестирование для характеристик, присущих аналогичным продуктам, так и тестирование для характеристик, указанных в описании продукта. Указания охватывают тестирование путем проверки документов и тестирование программ и данных по принципу «черного ящика».

Работы по тестированию

- Описание продукта, документация пользователя, программы и любые данные, поставляемые как части пакета программ, должны быть протестированы на выполнение ими формулировок и требований.

Протоколы тестирования

- план тестирования или технические требования (спецификацию) к тестированию, содержащие контрольные примеры (для каждого контрольного примера указаны ею цели);
- все результаты, связанные с контрольными примерами, включая все ошибки, выявленные при выполнении теста;
- штат персонала, вовлеченного в тестирование.

Отчет о тестировании

1.Обозначение продукта,

2 Вычислительные системы, использованные при тестировании (технические средства, программные средства и их конфигурация).

3. Использованные документы (включая их обозначения).

4. Результаты тестирования описания продукта, документации пользователя, программ и данных.

5. Перечень несоответствий требованиям.

6.Перечень несоответствий рекомендациям либо перечень не учтенных в продукте рекомендаций, либо формулировка того, что продукт не был протестирован на соответствие рекомендациям.

7. Дата окончания тестирования.

Дополнительное тестирование

- все измененные части документов, функций и данных должны быть протестированы как новый продукт;
- все неизмененные части, на которые могут влиять измененные части или изменения в необходимой системе (в соответствии с опытной оценкой тестировщика), должны быть протестированы как новый продукт;
- все другие части должны быть по крайней мере выборочно протестированы.

стандарт 1EEE 1209-1992

В стандарте 1EEE 1209-1992 сформулированы *общие требования к функциям средств автоматизации тестирования, ,* входящим в CASE-средства, которые должны обеспечивать:

- определение тестов — реализацию процесса тестирования пользователем: ввод тестовых наборов, генерацию тестовых наборов;
- генерацию тестовых данных, ввод ожидаемых, эталонных результатов, генерацию ожидаемых результатов;
- выполнение участка тестируемой программы между контрольными точками
- управление тестами и участком программы
- регрессионное тестирование с возвратом от более сложных тестов к простым
- анализ тестовых результатов
- анализ покрытия тестами исходного кода для обнаружения операторов (элементов текста программы, выражающих законченные действия
- анализ производительности программы
- верификацию условий или исключительных ситуаций во время выполнения теста
- моделирование среды - поддержку процесса тестирования с помощью модели

Организация и этапы тестирования при испытаниях надежности сложных программных средств

Цели и этапы испытаний надежности комплексов программ.

Основные этапы тестирования и испытаний комплекса программи его компонентов представлены на рис.

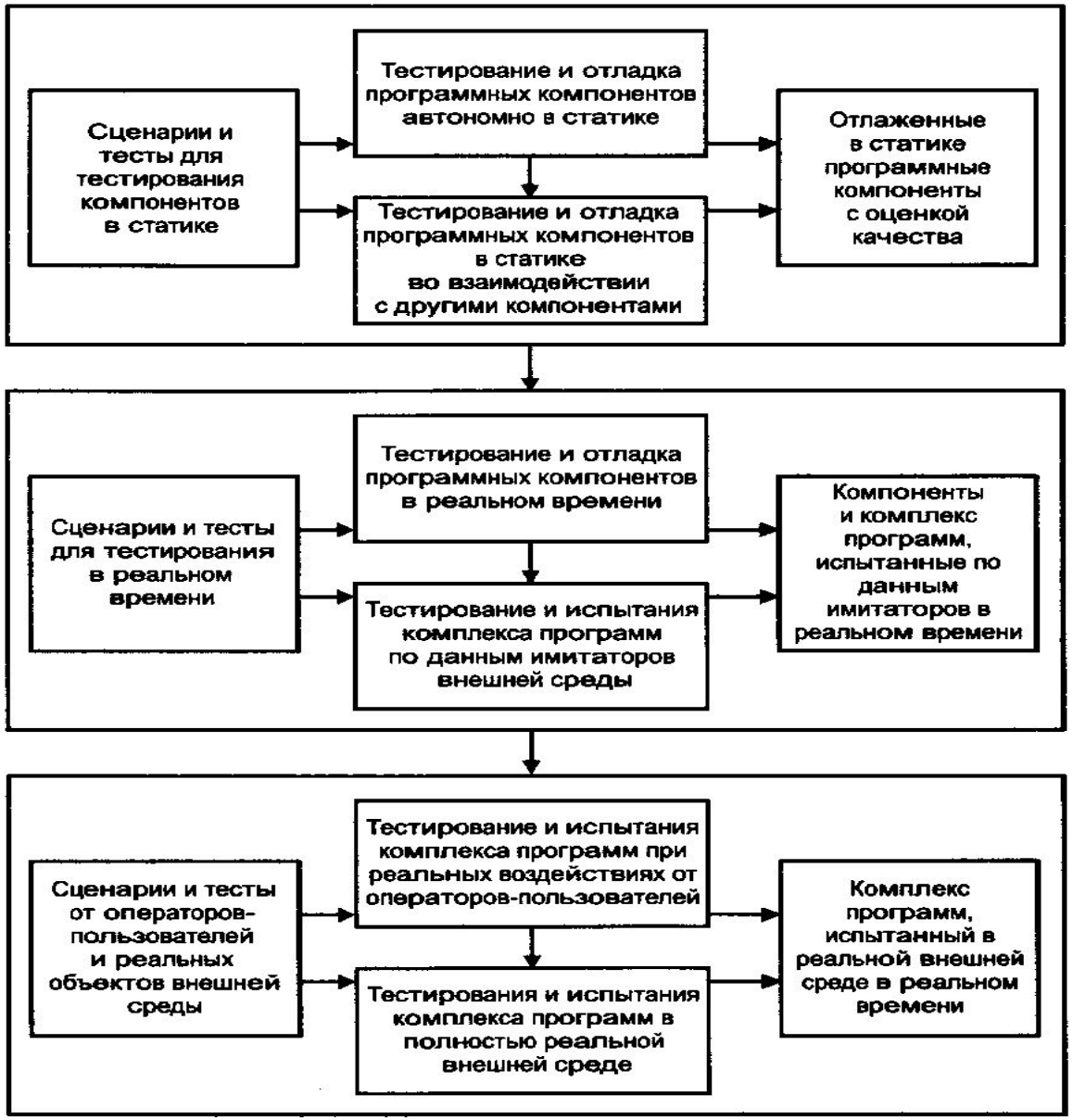


Рис. 5.4. Схема этапов тестирования и испытаний сложных комплексов программ

Для каждого этапа на рисунке представлены основные исходные данные и результаты тестирования и испытаний. При тестировании, отладке и испытаниях *корректности* ***компонентов комплексов программ выделены следующие этапы:***

- комплексирование модулей и отладка автономных групп программ в статике без взаимодействия с другими компонентами и, возможно, без подключения к операционной системе реального времени;
- тестирование и отладка групп программ в статике с учетом взаимодействия с некоторыми другими важнейшими компонентами и с базой данных;
- тестирование и отладка отдельных программных компонентов в реальном времени во взаимодействии с другими функциональными компонентами и с основными компонентами операционной системы и базы данных.

стадии комплексного тестирования и испытаний ПС в реальном времени;

- по данным моделирующего стенда или генераторов тестов, имитирующих отдельные объекты внешней среды;
- с имитаторами отдельных объектов внешней среды и с реальными воздействиями от операторов-пользователей;
- в полностью адекватной реальной или имитированной внешней среде и с реальными воздействиями от операторов-пользователей.

Первая группа — это работы по методическому обеспечению тестирования и по созданию средств автоматизированной генерации тестов. *Вторая группа* работ должна обеспечивать возможность обработки результатов тестирования и оценки достигнутых показателей качества функционирования программ.

Методика тестирования при испытаниях надежности сложных программных среда!

- **Тестирование и отладка программных компонентов в реальном времени**
- **Тестирование и испытания комплекса программ по данным имитаторов внешней среды**
- **Тестирование и испытания надежности комплекса программ при воздействиях операторов-пользователей**
- **Испытания комплекса программ в реальной внешней среде**

Тестирование программного обеспечения

Разработано множество стандартов и методик поддержки стадий ЖЦ ПО, например **стандарты 13О 9000 и 15О 9001**, разработанные Международной организацией по стандартизации (18О).

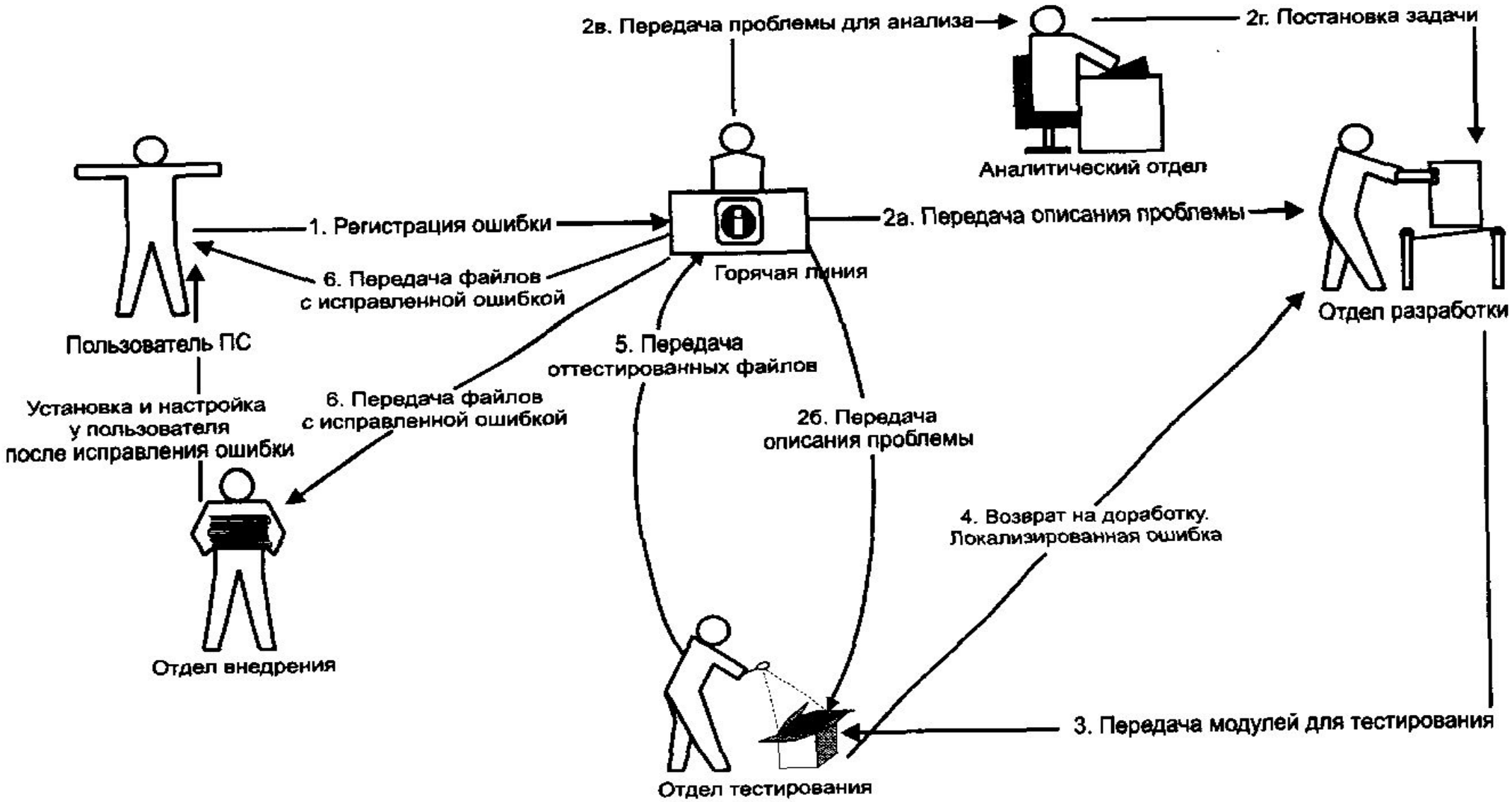


Рис. 5.5. Схема технологического процесса исправления ошибки ПС при обнаружении ее пользователем