

Современные
направления проверки
правильности
программ

- К методам проектирования ПС относятся структурные, объектно-ориентированные и др.
- Их основу составляют теоретические, инструментальные и прикладные средства, которые влияют на процесс тестирования.
- **Теоретические средства** определяют процесс программирования и тестирования программного продукта. К ним относятся методы верификации и доказательства правильности спецификации программ
- **Инструментальные средства** - это способы поддержки кодирования и тестирования (компиляторы, генераторы программ, отладчики и др.), а также организационные средства планирования и отбора тестов для программ, которые обеспечивают обработку текста на ЯП и подготовку для них соответствующих тестов.

- При проверке правильности программ и систем рассматриваются процессы **верификации, валидации и тестирования ПС**, которые регламентированы в стандарте ISO/IEC 12207 [7.7] жизненного цикла ПО. В некоторой зарубежной литературе процессы верификации и тестирования отождествляются.
- **Тестирование** - это процесс обнаружения ошибок в ПО путем исполнения выходного кода ПС на тестовых данных, сбора рабочих характеристик в динамике выполнения в конкретной операционной среде, выявления различных ошибок, дефектов, отказов и изъянов, вызванных нерегулярными и аномальными ситуациями или аварийным прекращением работы ПО.

Процессы ЖЦ верификация и валидация программ

- Верификация и валидация, как методы, обеспечивают соответственно проверку и анализ правильности выполнения заданных функций и соответствия ПО требованиям заказчика, а также заданным спецификациям.
- Они представлены в стандартах как самостоятельные процессы ЖЦ и используются, начиная от этапа анализа требований и кончая проверкой правильности функционирования программного кода на заключительном этапе, а именно, тестировании.

их трактовку в стандартном представлении

- **Процесс верификации.** Цель процесса - убедиться, что каждый программный продукт (и/или сервис) проекта отражает согласованные требования к спецификации.
- Этот процесс основывается:
- на стратегии и критериях верификации применительно ко всем рабочим программным продуктам;
- на выполнении действий стандарта по верификации;
- на устранении недостатков, обнаруженных в программных (рабочих и промежуточных) продуктах;
- Процесс верификации может проводиться исполнителем программы или другим сотрудником той же организации, или сотрудником другой организации, например заказчиком. Этот процесс включает в себя действия по его внедрению и выполнению.

Процесс валидации

- **Цель процесса** - убедиться, что специфические требования для программного продукта выполнены, и осуществляется это с помощью:
- разработанной стратегии и критериев валидации для всех рабочих продуктов;
- оговоренных действий по проведению валидации;
- демонстрации соответствия разработанных программных продуктов требованиям заказчика и правилам их использования;
- согласования с заказчиком полученных результатов валидации.
- Таким образом, основные задачи процессов верификации и валидации состоят в том, чтобы проверить и подтвердить, что конечный программный продукт отвечает назначению и удовлетворяет требованиям заказчика. Эти процессы взаимосвязаны и определяются, как правило, одним общим термином "верификация и валидация" или "Verification and Validation" (V&V)

Чем отличается валидация от верификации

- Стандарт ИСО 9000:2000 определяет эти термины следующим образом:
- "Верификация - подтверждение на основе представления объективных свидетельств того, что установленные требования были выполнены".
- "Валидация - подтверждение на основе представления объективных свидетельств того, что требования, предназначенные для конкретного использования или применения, выполнены".

- Уже перевод с английского этих терминов дает определенную пищу для понимания разницы: verification - проверка, validation - придание законной силы.
- Чтобы было проще понять, сразу приведу пример типичной верификации: тестирование программы или проведение испытания оборудования. Имея определенные требования на руках, мы проводим испытание продукта и фиксируем, соблюдены ли требования. Результат верификации - это ответ на вопрос "Соответствует ли продукт требованиям?".
- Но далеко не всегда продукт, соответствующий установленным требованиям, можно применять в конкретной ситуации. Например, лекарство прошло все положенные испытания и поступило в продажу. Значит ли это что оно может быть применено каким-то конкретным больным? Нет, т.к. каждый пациент имеет свои особенности и конкретно для этого лекарство может быть губительным, т.е. кто-то (врач) должен подтвердить: да, этому больному можно принимать это лекарство. То есть врач должен выполнить валидацию: придать законную силу конкретному применению.

- верификация - проводится практически всегда, выполняется методом проверки (сличения) характеристик продукции с заданными требованиями, результатом является вывод о соответствии (или несоответствии) спецификации,
- валидация - проводится при необходимости, выполняется методом анализа заданных условий применения и оценки соответствия характеристик продукции этим требованиям, результатом является вывод о возможности применения продукции для конкретных условий.

Тестирование программ

- тестирование предполагает выполнение программы и получение конкретных результатов выполнения тестов
- Исторически первым видом тестирования была **отладка**
- **Отладка** - это проверка описания программного объекта на качество с целью обнаружения в нем ошибок и последующее их устранение. Ошибки обнаруживаются компиляторами при их синтаксическом контроле.
- После этого проводится верификация по проверке правильности кода и валидация по проверке соответствия продукта заданным требованиям.
- **Целью тестирования** - проверка работы реализованных функций в соответствии с их спецификацией. На основе внешних спецификаций функций и проектной информации на процессах ЖЦ создаются функциональные тесты, с помощью которых проводится тестирование с учетом требований, сформулированных на этапе анализа предметной области.
- Методы функционального тестирования подразделяются на **статические и динамические**.

Статические методы тестирования

- Статические методы используются при проведении инспекций и рассмотрении спецификаций компонентов без их выполнения. Техника статического анализа заключается в методическом просмотре (или обзоре) и анализе структуры программ, а также в доказательстве их правильности. Статический анализ направлен на анализ документов, разработанных на всех этапах ЖЦ и заключается в инспекции исходного кода и сквозного контроля программы.

Динамические методы тестирования

- Динамические методы тестирования используются в процессе выполнения программ. Методы, в которых программы рассматриваются как "**черный ящик**" (используется информация о решаемой задаче), и методы, в которых программа рассматривается как "**белый ящик**" (используется структура программы).
- Цель динамического тестирования программ по принципу "черного ящика" - выявление одним тестом максимального числа ошибок с использованием небольшого подмножества возможных входных данных.
- Метод "белого ящика" позволяет исследовать внутреннюю структуру программы, причем обнаружение всех ошибок в программе является критерием исчерпывающего тестирования маршрутов потоков (графа) передач управления.

Функциональное тестирование

- Цель функционального тестирования - обнаружение несоответствий между реальным поведением реализованных функций и ожидаемым поведением в соответствии со спецификацией и исходными требованиями. Функциональные тесты должны охватывать все реализованные функции с учетом наиболее вероятных типов ошибок. Тестовые сценарии, объединяющие отдельные тесты, ориентированы на проверку качества решения функциональных задач.

В задачи функционального тестирования входят:

- идентификация множества функциональных требований;
- идентификация внешних функций и построение последовательностей функций в соответствии с их использованием в ПС;- идентификация множества входных данных каждой функции и определение областей их изменения; построение тестовых наборов и сценариев тестирования функций; выявление и представление всех функциональных требований с помощью тестовых наборов и проведение тестирования ошибок в программе и при взаимодействии со средой.

Ошибки в разработке программ

- **Отказ** (failure) - это отклонение программы от функционирования или невозможность программы выполнять функции, определенные требованиями и ограничениями, рассматривается как событие, способствует переходу программы в неработоспособное состояние из-за ошибок, скрытые в ней дефекты или сбои в среде функционирования
- Отказ может быть по следующим причинам:
- - Ошибочная спецификация или пропущено требование, которое означает, что спецификация точно не отражает того, что предполагал пользователь;
- - Спецификация может содержать в себе требование, которое невозможно выполнить на данной аппаратуре и программном обеспечении;
- - Проект программы может содержать в себе ошибки (например, база данных спроектирована без средств защиты от несанкционированного доступа пользователя, а нужна защита);
- - Программа может быть неправильной, то есть она выполняет несвойственный алгоритм или он реализован не полностью.

Общие классы ошибок.

- - Логические и сбои;
 -
 - - Ошибки вычислений и времени выполнения;
 -
 - - Ошибки ввода-вывода и манипулирования данными;
 -
 - - Ошибки интерфейсов;
 -
 - - Ошибки объема данных и др..
-
- **Логические ошибки** - следствие нарушения логики алгоритма, внутренней несогласованности переменных и операторов, а также языковых правил программирования.
Сбои - следствие неправильно определенных функций, нарушение порядка их применения или отсутствия полноты их реализации и т.д.

Общие классы ошибок.

- **Ошибки вычислений** возникают из-за неточности исходных данных и реализуемых формул, погрешностей методов, неправильного применения операций вычислений. Ошибки времени выполнения связанные с отсутствием необходимой скорости обработки запросов, или времени выполнения или восстановления программы.
-
- **Ошибки ввода-вывода и манипулирования данными** является следствием некачественной подготовки данных для выполнения программы, сбоев при занесении их в базу данных или при выборке из нее.
-
- **Ошибки интерфейса** относятся к ошибкам взаимосвязи отдельных элементов друг с другом, что проявляется при передаче данных между ними, а также при взаимодействии со средой функционирования.
-
- **Ошибки объема** относятся к данным и является следствием того, что реализованные методы доступа и размеры баз данных не удовлетворяют реальные объемы информации системы или интенсивности их обработки.

- Проявляются ошибки в программах по-разному.
- Так, при работе с БД возникают ошибки представления и манипулирования данными, логические ошибки в задании прикладных процедур обработки данных и др.
- В программах вычислительного характера преобладают ошибки вычислений, а в программах управления и обработки - логические и сбои.
- В ПС, состоящий из многих разноязычных программ, реализующих различные функции, могут содержаться ошибки различных типов. Ошибки интерфейсов и нарушения объема характерные для любого типа ПС.

Процентное соотношение ошибок при разработке ПС



Связь ошибки с отказом.

- Наличие ошибки в программе, как правило, приводит к отказу ПС при его функционировании. Для анализа причинно-следственных связей «ошибка-отказ» выполняются следующие действия:
- - Идентификация ошибок в технологиях проектирования и программирования;
- - Взаимосвязь ошибок процесса проектирования и ошибок, допускаемых человеком;
- - Классификация отказов, ошибок и возможных ошибок, а также дефектов на каждом процессе разработки;
- - Сопоставление ошибок человека, допускаемых на определенном процессе разработки, и дефектов в объекте, как последствий ошибок спецификации проекта, моделей программ;
- - Проверка и защита от ошибок на всех процессах ЖЦ, а также выявление дефектов на каждом процессе разработки;
- - Сопоставление дефектов и отказов в ПС для разработки системы взаимосвязей и методики локализации, сбора и анализа информации об отказах и дефекты;
- - Разработка подходов к процессам документирования и сопровождения ПС.

Классификацию типов отказов

- Конечная цель причинно-следственных связей «ошибка-отказ» заключается в определении методов и средств тестирования и выявления ошибок определенных классов, а также критериев завершения тестирования на множестве наборов данных в определении путей совершенствования организации процесса разработки, тестирования и сопровождения ПС.

Приведем следующую классификацию типов отказов:

- Аппаратные;
- Информационные, вызваны ошибками во входных данных и передачи данных по каналам связи, а также при сбоях устройств ввода, как следствие аппаратных отказов;
- Программные, при наличии ошибок в компонентах системы;
- Эргономичные, вызваны ошибками оператора при его взаимодействии с компьютером (это отказ - вторичный отказ, может привести к информационному или функциональному отказам).

- Исследование фирм IBM показали, чем позже обнаруживается ошибка в программе, тем дороже стоит ее исправление, эта зависимость близка к экспоненциальной. Так, военно-воздушные силы США оценили стоимость разработки одной инструкции в 75 долларов, а стоимость сопровождения составляет около 4000 долларов.



Классификация тестов проверки



- В зависимости от задач, которые ставятся перед тестированием программ, составляются тесты проверки промежуточных результатов проектирования элементов на этапах ЖЦ, а также создаются тесты испытаний окончательного кода системы.
- Тесты интегрированной системы. Тесты для проверки отдельных элементов системы и тесты интегрированной системы имеют общие и отличительные черты.
- Так, на рис. в качестве примера приведена схема интеграции готовых оттестированных элементов. В ней заданы связи между разными уровнями тестирования интегрируемой ПС.

Интегрированное тестирование

КОМПОНЕНТ

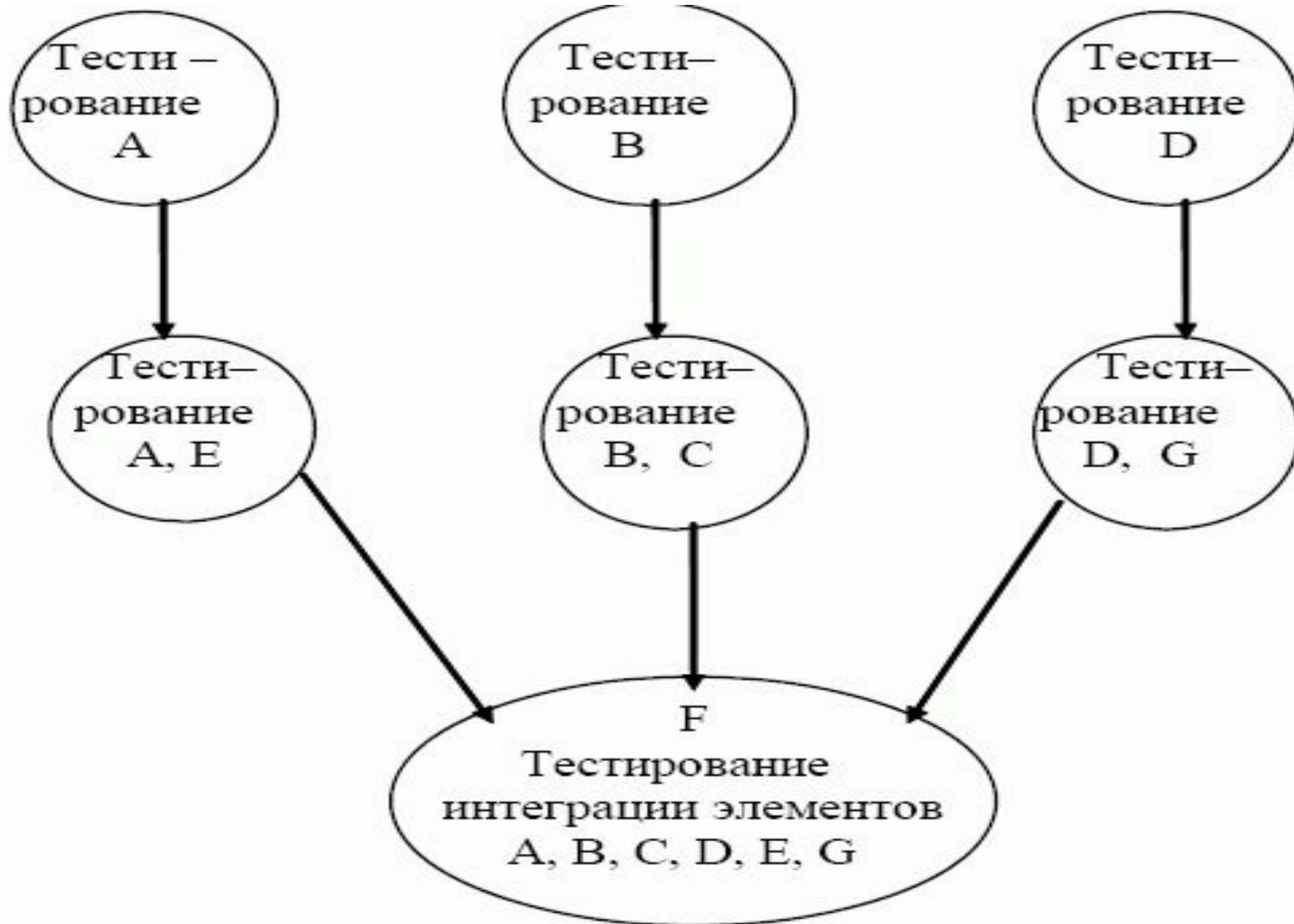


Схема ответственности инженера-тестировщика

- Для проведения тестирования тестовые инженеры предлагают процедуры тестирования (Test Procedures), содержащих валидацию объектов и верификацию тестовых сценариев согласно плана-графика. Оценка тестов (Test Evaluation) заключается в оценке результатов тестирования, степени покрытия программ сценариями и статуса полученных ошибок.
- Продавец интегрированной системы проверяет интерфейсы и дает оценку выполнения соответствующих системных тестов, а затем анализирует результаты тестирования.

Ответственности инженера-тестировщика



Управление процессом тестирования

- Все способы тестирования ПС объединяются базой данных, где помещаются результаты тестирования системы. В ней содержатся все компоненты, тестовые контрольные данные, результаты тестирования и информация о документировании процесса тестирования.
- База данных проекта поддерживается специальными инструментальными средствами типа CASE, которые обеспечивают ведение анализа ПрО, сборку данных об их объектах, потоках данных и тому подобное. База данных проекта хранит также начальные и эталонные данные, которые используются для сопоставления данных, накопленных в базе, с данными, которые получены в процессе тестирования системы.
- Результаты данного процесса отображаются на дисплее, например, в графической форме (пути прохождения по графу программы), в виде диаграмм UML, данных об отказах и ошибках или конкретных значений исходных параметров программы. Эти данные анализируются разработчиками для формулирования выводов о направлениях дальнейшей проверки правильности программы или их завершении.

- Результаты данного процесса отображаются на дисплее, например, в графической форме (пути прохождения по графу программы), в виде диаграмм UML, данных об отказах и ошибках или конкретных значений исходных параметров программы. Эти данные анализируются разработчиками для формулирования выводов о направлениях дальнейшей проверки правильности программы или их завершении.
- Документирование результатов тестирования в соответствии с действующим стандартом ANSI/IEEE 829 включает:
 - описание задач, назначение и содержание ПС, а также перечень функций в соответствии с требованиями заказчика;
 - технологии разработки системы;
 - планов тестирования различных объектов, необходимых ресурсов, соответствующих специалистов для проведения тестирования и технологических способов;
 - тестов, контрольных примеров, критериев и ограничений, оценки результатов программного продукта, а также процесса тестирования;
 - учета процесса тестирования, составление отчетов об аномальных событиях, отказах и дефектах в итоговом документе системы.

Виды вычислений характеристик процесса по методам планирования и управления

- При тестировании выполняются различные виды вычислений характеристик процесса по методам планирования и управления.
- 1. Расчет длительности выполнения функций путем сбора средних показателей скорости выполнения операторов без выполнения программы на машине. Выявляются компоненты, требующие длительного времени выполнения в реальной среде.
- 2. Управление выполнением тестирования путем подбора тестов проверки, их выполнения, селекции результатов тестирования и сопоставления их с эталонными значениями. Результаты данного процесса отображаются на дисплее, например, ветви выполнения в графической форме, данные об отказах и ошибках или конкретные значения выходных параметров программы. Эти данные анализируются разработчиками для формулирования выводов о направлениях дальнейшей проверки правильности программы или их завершении.
- 3. Планирование тестирования предназначено для распределения сроков работ по тестированию, распределения менеджера по отдельным видам работ и сдачи ими тестов проверки системы. Определяется стратегия и пути тестирования. В диалоге запрашиваются данные о реальных значениях процесса выполнения системы, структуры ветвления вершин графа и параметрах циклов.

Виды вычислений характеристик процесса по методам планирования и управления

- 3. Планирование тестирования предназначено для распределения сроков работ по тестированию, распределения менеджера по отдельным видам работ и сдачи ими тестов проверки системы. Определяется стратегия и пути тестирования. В диалоге запрашиваются данные о реальных значениях процесса выполнения системы, структуры ветвления вершин графа и параметрах циклов.
 - Проверенные циклы, как правило, изымаются из путей выполнения программы.
 - При планировании путей выполнения создаются соответствующие тесты, критерии и входные значения.
- 4. Результаты тестирования документируются согласно действующему стандарту ANSI / IEEE 829 и включают в себя:
 - - Описание задач, назначение и содержание ВС, а также перечень функций в соответствии с требованиями заказчика;
 - - Технологию разработки системы;
 - - Планы тестирования различных объектов, соответствующих технологическим приемам проведения тестирования;
 - - Тесты, контрольные примеры, критерии и ограничения, методику оценки результатов выполнения программного продукта в процессе тестирования;
 - - Учет процесса тестирования, составления отчетов о аномальных событиях, отказах и дефектах в итоговом документе системы.

Контрольные вопросы и задания

- 1. Назовите формальные методы проверки правильности программ.
- 2. Определите формальную спецификацию.
- 3. Назовите цели процессов верификации и валидации программ.
- 4. Определите процесс тестирования и методы тестирования.
- 5. Объясните значение терминов «черный ящик» и «белая ящик»б. Назовите объекты тестирования и подходы к их тестирования.
- 6. Какая существует классификация типов ошибок и тестов проверки программ.
- 7. Назовите сущность инфраструктуры организации работ по тестированию?