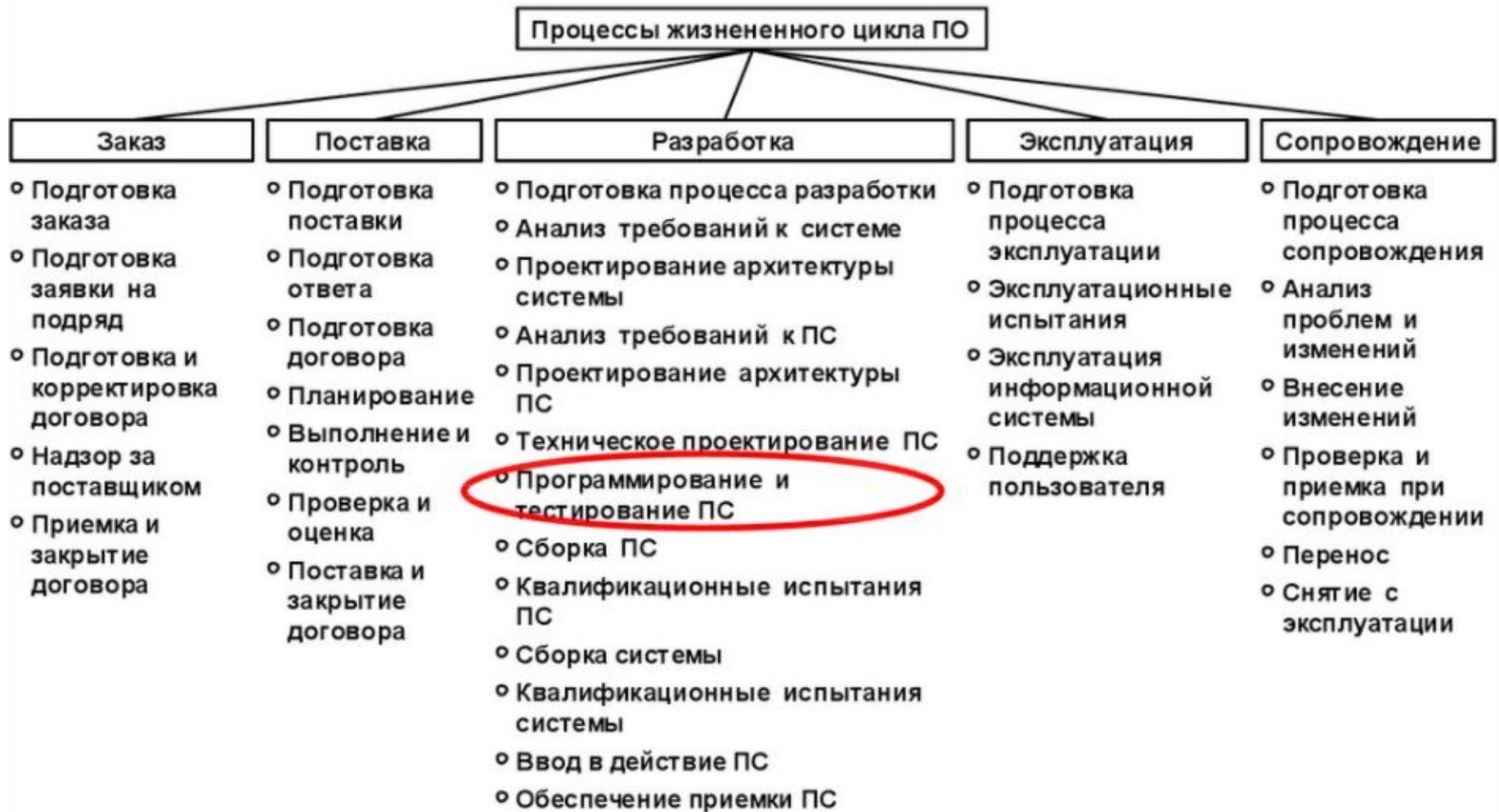


Тема 10. Тестирование и отладка программного средства

Процессы жизненного цикла ПО согласно МС ISO/IEC 12207:95 (ГОСТ Р ИСО/МЭК 12207:99)



Состав задач тестирования ПС

согласно МС ISO/IEC 12207:95 (ГОСТ Р ИСО/МЭК 12207:99)

Разработка процедуры испытаний (тестирования) и данных для тестирования каждого программного модуля и базы данных

Тестирование каждого программного модуля и базы данных

Уточнение документации пользователя (при необходимости)

Уточнение требований к тестированию

Оценка программных модулей и результатов их тестирования по следующим критериям:

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

Часть 1. Основные понятия

Отладка ПС – это деятельность, направленная на обнаружение и исправление ошибок в ПС.

Тестирование ПС – это процесс выполнения программ на некотором **наборе данных**, для которого заранее известен результат применения или известны правила поведения этих программ. **Набор данных** называется тестовым.

Отладка = Тестирование + Поиск ошибок (в программах и документации) + **Редактирование** (с целью устранения ошибок).

См. рисунок

Основные определения тестирования и отладки ПС



Основные определения тестирования и отладки ПС



Тестирование всегда выполняется с помощью некоторого набора данных, который называется тестовым или тестом



При этом ...

Хорошим считают тест с высокой вероятностью обнаружения еще не раскрытой ошибки

Успешным называют тест, который обнаруживает до сих пор не раскрытую ошибку

Основные определения тестирования и отладки ПС



ВАЖНО ПОМНИТЬ!



Тестирование не может показать отсутствия дефектов

- оно может показывать только присутствие дефектов

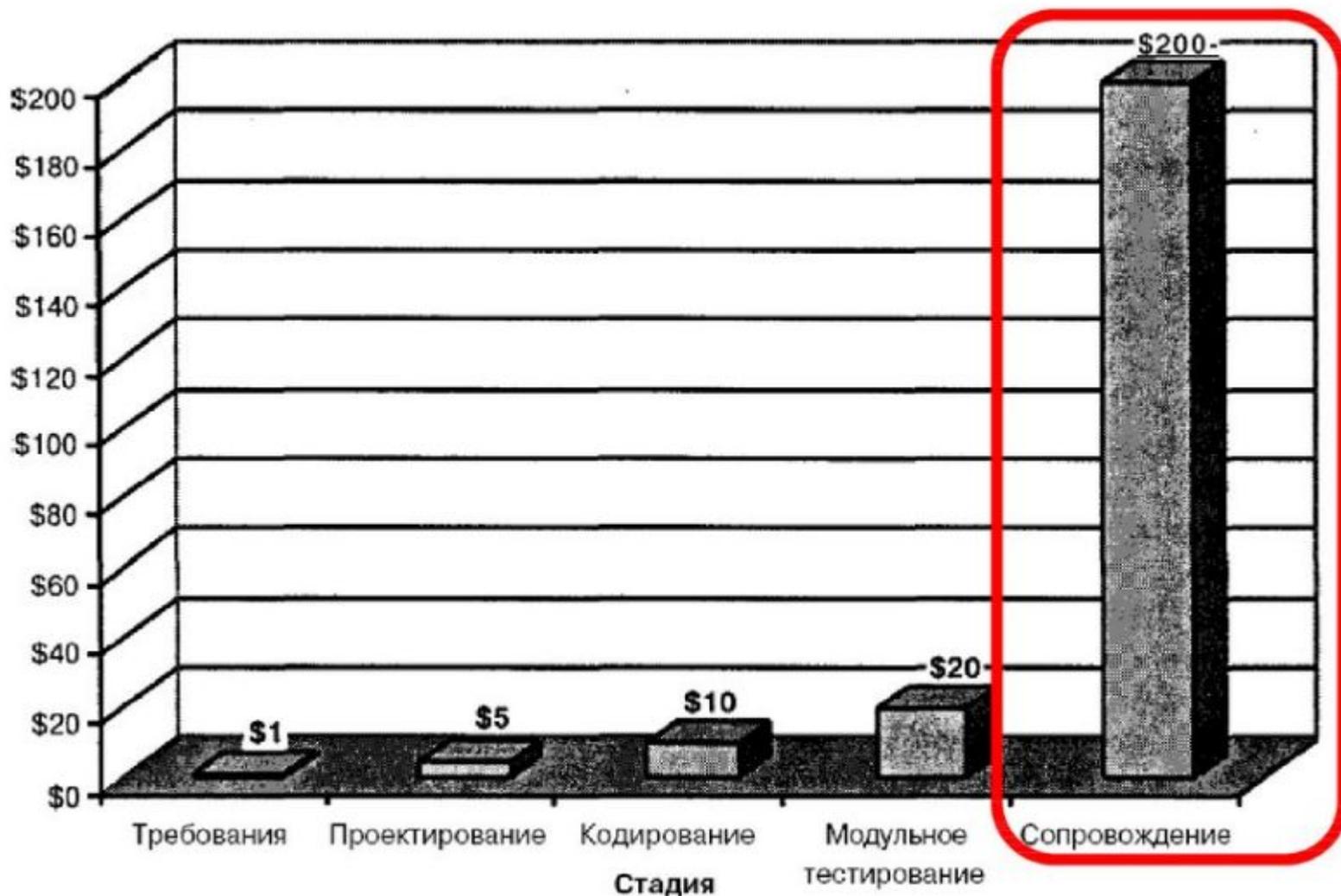


Следовательно ...

1 Если тесты не обнаруживают ошибок, необходимо проверить достаточно ли продуманы тесты и что в ПО нет скрытых ошибок

2 Ошибки не выявленные с помощью тестов будут обнаруживаться на стадии эксплуатации ПО, когда стоимость их устранения возрастет в 60 – 100 раз

Стоимость устранения ошибок в программном обеспечении на разных стадиях его жизненного цикла



Виды тестирования ПО

Виды тестирования ПО

По объекту тестирования

- Функциональное тестирование
- Тестирование производит.
- Юзабилити-тестирование
- Тестирование интерфейса пользователя
- Тестирование безопасности
- Тестирование локализации
- Тестирование совместимости

По знанию системы

- Тестирование чёрного ящика
- Тестирование белого ящика
- Тестирование серого ящика

По степени автоматизации

- Ручное тестирование
- Автомат. тестирование
- Полуавтомат. тестирование

По степени изолированности компонентов

- Компонентное (модульное) тестирование
- Интеграционное тестирование
- Системное тестирование

По времени проведения тестирования

- Альфа-тестирование
- Бета-тестирование

По признаку позитивности сценариев

- Позитивное тестирование
- Негативное тестирование

По степени подготовленности к тестированию

- Тестирование по документации
- Интуитивное тестирование

Принципы и виды отладки ПС

Кратко (слайды 10-12), подробнее на рисунках (слайды 14-23)

При тестировании ПС возникает две задачи:

1. Подготовка набора тестов

(чем дольше продолжается процесс тестирования, тем большей становится стоимость ПС)

2. Определение момента окончания отладки ПС.

(признак возможности окончания отладки является полнота охвата пропущенными через ПС тестами множества различных ситуаций, возникающих при выполнении программ и относительно редкое проявление ошибок в ПС на последнем отрезке процесса тестирования.

Проектирование тестов

А) левый крайний подход

Тесты проектируются только на основании изучения спецификаций ПС, тестирование осуществляется путем перебора всех наборов входных данных, что практически неосуществимо.

Б) крайний правый подход

Тесты проектируются на основании изучения текстов программ с целью протестировать все пути выполнения (много) каждой программы ПС, такое тестирование также практически неосуществимо.

В) оптимальная стратегия проектирования тестов

Включает проектирование значительной части тестов по спецификациям, но она требует также и проектирования некоторых тестов и по текстам программ.

Принципы оптимальной стратегии проектирование тестов

Вариант 1

- На каждую используемую функцию или возможность предусматривается хотя бы один тест.
- На каждую область и на каждую границу изменения какой-либо входной величины – хотя бы один тест.
- На каждую особую исключительную ситуацию – хотя бы один тест.

Вариант 2

Оптимальная стратегия базируется на принципе: каждая команда каждой программы ПС должна проработать хотя бы на одном тесте.

См. рисунки

Принципы и виды отладки программного обеспечения

**Успех отладки ПС в значительной степени
предопределяет рациональная организация тестирования**



Для рациональной организации отладки ПО
необходимо решить следующие задачи ...

1 Подготовить такой набор тестов, чтобы обнаружить в ПС по возможности большее число ошибок.

2 Определить момент окончания отладки ПС (или отдельной его компоненты)



Для оптимизации набора тестов
необходимо

1 Заранее планировать этот набор

2 Использовать рациональную стратегию планирования (проектирования) тестов

Принципы и виды отладки программного обеспечения

Успех отладки ПС в значительной степени
предопределяет рациональная организация тестирования



Для рациональной организации отладки ПО
необходимо решить следующие задачи ...

1 Подготовить такой набор тестов, чтобы обнаружить в ПС по возможности большее число ошибок.

2 Определить момент окончания отладки ПС (или отдельной его компоненты)

Признаки
окончания
отладки



1 Полнота охвата выполненными тестами множества различных ситуаций, возникающих при выполнении ПС

2 Относительно редкое проявление ошибок в ПС на последнем отрезке процесса тестирования



При этом, важно помнить!

Чем дольше продолжается процесс тестирования
(и отладки в целом), тем большей становится стоимость ПС.

Успех отладки ПС в значительной степени предопределяет рациональная организация тестирования

Для рациональной организации отладки ПО
необходимо решить следующие задачи ...

1 Подготовить такой набор тестов, чтобы обнаружить в ПС по возможности большее число ошибок.

2 Определить момент окончания отладки ПС (или отдельной его компоненты)

Для оптимизации набора тестов
необходимо

1 Заранее планировать этот набор

2 Использовать рациональную стратегию планирования (проектирования) тестов

Выбор оптимальной стратегии находится между двумя крайними подходами

Тестирование
"Черного ящика"

Тестирование по отношению к спецификациям

Тестирование
"Белого ящика"

Тестирование по отношению к текстам программ

Оптимальная стратегия

Хотя бы один тест на каждую:
- используемую функцию;
- область и границу изменения какой-либо входной величины;
- особую (исключительную) ситуацию.

Хотя бы один тест на каждую команду в каждом программном модуле ПС.

Планировать набор тестов можно после получения требований к ПС, уточняя его на каждой стадии жизненного цикла ПС



Успех отладки ПС в значительной степени
предопределяет рациональная организация тестирования

Для рациональной организации отладки ПО
необходимо решить следующие задачи ...

1 Подготовить такой набор
тестов, чтобы обнаружить
в ПС по возможности
большее число ошибок.

2 Определить момент
окончания отладки
ПС (или отдельной
его компоненты)

Для оптимизации набора тестов
необходимо

1 Заранее планировать этот набор

2 Использовать рациональную стратегию
планирования (проектирования) тестов

Выбор оптимальной стратегии находится
между двумя крайними подходами

Тестирование
"Черного ящика"

Тестирование
по отношению
к спецификациям

Тестирование
"Белого ящика"

Тестирование
по отношению
к текстам программ

Оптимальная
стратегия

Хотя бы один тест на каждую:
- используемую функцию;
- область и границу изменения
какой-либо входной величины;
- особую (исключительную) ситуацию.

Хотя бы один тест на каждую
команду в каждом
программном модуле ПС.

Тесты проектируются
только на основании
изучения
спецификации
ПС. Строение модулей
при этом никак
не учитывается,
т.е. они
рассматриваются
как "Черные ящики"

Тесты проектируются
на основании изучения
текстов программ
с целью протестировать
все пути выполнения
каждой программ ПС.
Т.е. программные
модули
рассматриваются,
как "Белые ящики".

Тестирование
"Черного ящика"

Тестирование
"Белого ящика"

Тестирование
по отношению
к спецификациям

Тестирование
по отношению
к текстам программ

Оптимальная
стратегия

Хотя бы один тест на каждую:
- используемую функцию;
- область и границу изменения
какой-либо входной величины;
- особую (исключительную) ситуацию.

Хотя бы один тест на каждую
команду в каждом
программном модуле ПС.

Тесты проектируются только
на основании изучения спецификации ПС.

Строение модулей при этом никак не
учитывается, т.е. они рассматриваются
как **"Черные ящики"**

Тесты проектируются на основании
изучения текстов программ
с целью протестировать все пути
выполнения каждой программ ПС.
Т.е. программные модули
рассматриваются, как **"Белые ящики"**.

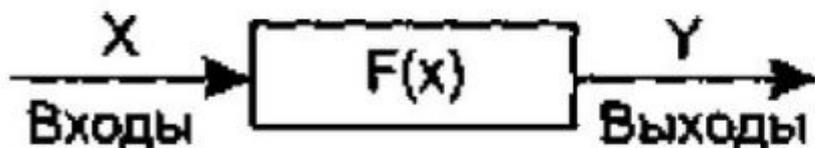
Подход требует
Полного перебора
всех наборов входных данных,
так как в противном случае
некоторые участки программ ПС
могут не работать при пропуске
любого теста, а это значит,
что содержащиеся в них ошибки
не будут проявляться. **Такое
тестирование практически
неосуществимо**

При этом
Если принять во внимание наличие
в программах циклов с переменным
числом повторений, то различных
путей выполнения программ ПС
может оказаться также чрезвычайно
много, так что их **тестирование
будет практически неосуществимо**

Тестирование ПС по отношению к спецификациям – тестирование “Черного ящика”



«Черный ящик»



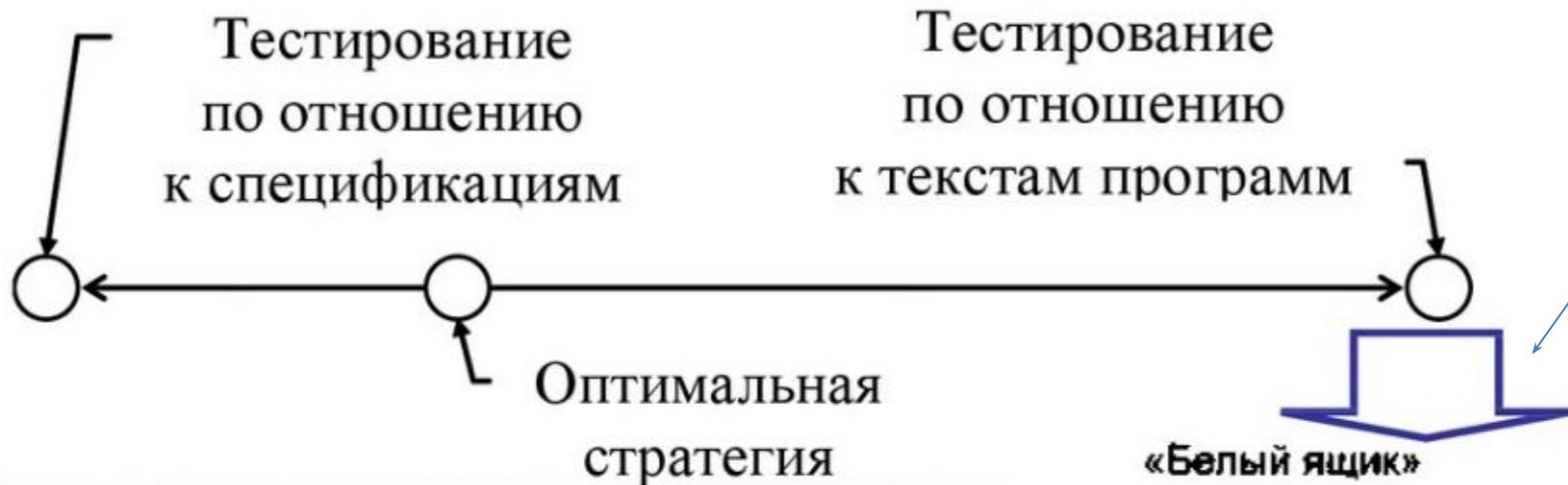
1 Известны: функции программы.

2 Исследуется: работа каждой функции на всей области определения.

При тестировании “Черного ящика” рассматриваются системные характеристики программ, игнорируется их внутренняя логическая структура.

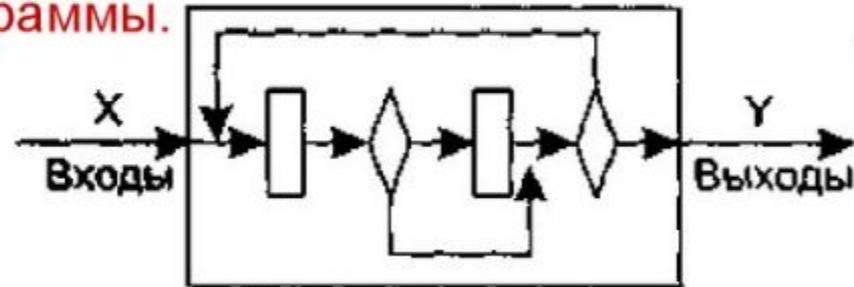
Полное или исчерпывающее тестирование, как правило, невозможно. Например, если в программе 10 входных величин и каждая принимает по 10 значений, то потребуется 10^{10} тестовых вариантов.

Подходы к тестированию программных средств



1 Известна: внутренняя структура программы.

2 Исследуются: внутренние элементы программы и связи между ними.



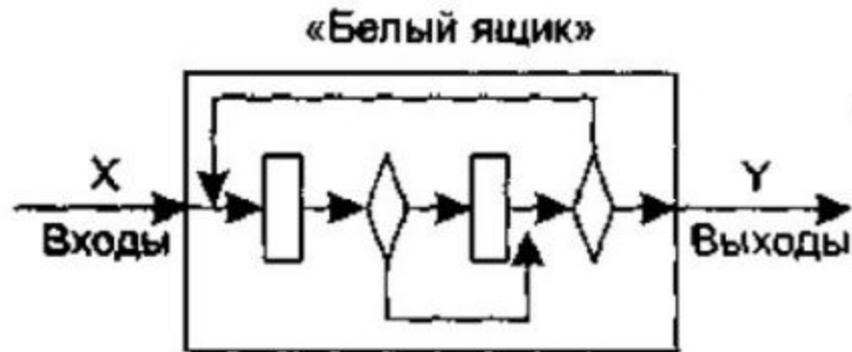
Объектом тестирования здесь является не внешнее, а внутреннее поведение программы.

Проверяется корректность построения всех элементов программы и правильность их взаимодействия друг с другом.

Обычно анализируются управляющие связи элементов, реже — информационные связи.

Исчерпывающее тестирование, как правило, невозможно.

Подходы к тестированию программных средств



Тестирование базового пути. В этом способе тесты разрабатываются для проверки базового множества путей – маршрутов в программе. Они гарантируют однократное выполнение каждого оператора программы при тестировании.

Тестирования условий позволяет строить тесты для проверки логических условий программы. При этом желательно обеспечить охват операторов из всех ветвей программы.

Тестирование ветвей и операторов отношений обнаруживает ошибки ветвления и операторов отношения.

Тестирования потоков данных. В данном способе анализу подвергается информационная структура программы.

Тестирование циклов. В этом способе особое внимание уделяется проверке правильности конструкций циклов.

Подходы к тестированию программных средств и выбор оптимальной стратегии тестирования

Тестирование
“Черного ящика”

Тестирование
“Белого ящика”

Тестирование
по отношению
к спецификациям

Тестирование
по отношению
к текстам программ



Хотя бы один тест на каждую:

- используемую функцию;
- область и границу изменения какой-либо входной величины;
- особую (исключительную) ситуацию.

Хотя бы один тест на каждую команду в каждом программном модуле ПС.

Оптимальная стратегия тестирования расположена внутри интервала между крайними подходами, но ближе к левому краю!

Подходы к тестированию программных средств и выбор оптимальной стратегии тестирования

Оптимальную стратегию проектирования тестов можно конкретизировать на основании следующего принципа:
– для каждого программного документа (включая тексты программ), входящего в состав ПС, должны проектироваться свои тесты с целью выявления в них ошибок



1 Результаты каждой стадии разработки ПО отражаются в соответствующих документах

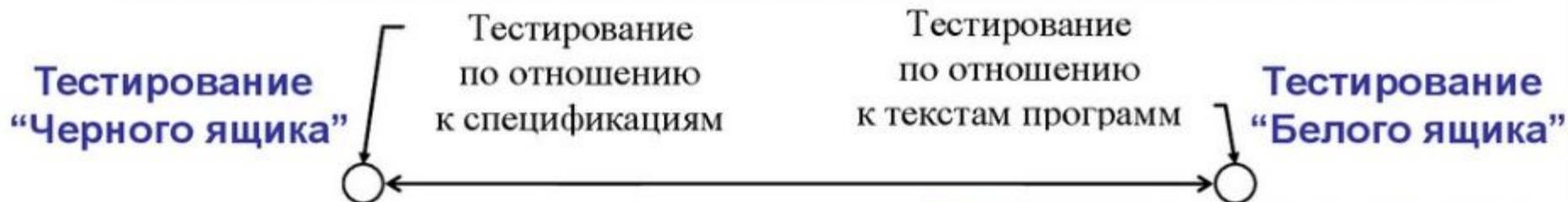
2 Набор и содержание тестов дополняется и уточняется на каждой стадии разработки ПО



Практическое задание

1 Опишите графически оптимальную стратегию тестирования Вашего программного средства.

2 Напишите комментарии к графику, поясняющие Ваш выбор.



Тесты проектируются только на основании изучения спецификации ПС.

Строение модулей при этом никак не учитывается, т.е. они рассматриваются как **"Черные ящики"**

Подход требует

Полного перебора всех наборов входных данных, так как в противном случае некоторые участки программ ПС могут не работать при пропуске любого теста, а это значит, что содержащиеся в них ошибки не будут проявляться.

Тесты проектируются на основании изучения текстов программ с целью протестировать все пути выполнения каждой программ ПС. Т.е. программные модули рассматриваются, как **"Белые ящики"**.

При этом

Если принять во внимание наличие в программах циклов с переменным числом повторений, то различных путей выполнения программ ПС может оказаться чрезвычайно много.