

# Методы тестирования

1. Тестирование «черного ящика»
2. Тестирование «белого ящика»

# Методы тестирования

Систематические методы тестирования делятся на методы, в которых программы рассматриваются как «**черный ящик**», и методы, в которых программа рассматривается как «**белый ящик**». Также можно выделить метод тестирования **серого ящика**, который сочетает в себе тестирование методом белого и черного ящика.

# Методы тестирования: чёрный ящик, белый ящик, серый ящик

## Методы тестирования

### Метод чёрного ящика

#### Black-Box-Verfahren

- Тестирование с точки зрения пользователя
- Внутренняя структура неизвестна
- Рассматриваем только вводимые данные и полученные результаты



### Метод серого ящика

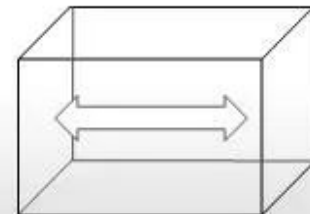
- Знания структуры частично использованы в тестировании
- Сами тесты написаны с точки зрения пользователя



### Метод белого ящика

#### Black-Box-Verfahren

- Тестирование с точки зрения разработчика
- Внутренняя структура известна
- Рассматриваем рабочие алгоритмы



# Тестирование «черного ящика»

Тестирование «черного ящика» — это метод тестирования функционального поведения объекта (программной системы) с точки зрения внешнего мира.

При тестировании методом «черного ящика» программа рассматривается как объект, внутренняя структура которого неизвестна.

Тестировщик вводит данные и анализирует результат, но он не знает, как именно работает программа. Подбирая тесты, специалист ищет интересные с его точки зрения входные данные и условия, которые могут привести к нестандартным результатам. Используются те представители каждого класса входных данных, при которых с наибольшей вероятностью могут проявиться ошибки тестируемой программы.

# Тестирование «черного ящика»

**Основная задача тестировщика для данного метода тестирования состоит в последовательной проверке соответствия поведения системы требованиям. Кроме того, тестировщик должен проверить работу системы в критических ситуациях — в случае подачи неверных входных значений. В идеальной ситуации все варианты критических ситуаций должны быть описаны в требованиях на систему, и тестировщику остается только придумывать конкретные проверки этих требований.**

# Тестирование «черного ящика»

Однако на самом деле в результате тестирования обычно выявляются **следующие проблемы системы:**

1. поведение системы не соответствует требованиям;
2. в ситуациях, не предусмотренных требованиями, система ведет себя неадекватно (при этом под неадекватным поведением может пониматься как полный крах системы, так и вообще любое поведение, не описанное в требованиях).

# Тестирование «черного ящика»

В первом случае обычно требуются **изменения программного кода**, гораздо реже — **изменения требований**.

Изменение требований может потребоваться из-за их противоречивости (несколько разных требований описывают разные модели поведения системы в одной и той же ситуации) или некорректности (требования не соответствуют действительности).

# Тестирование «черного ящика»

Во втором случае однозначно требуются изменения требований ввиду их неполноты — в требованиях явно не учтена ситуация, приводящая к неадекватному поведению системы.

**Методы «черного ящика» обеспечивают:**

- эквивалентное разбиение;
- анализ граничных значений;
- анализ причинно-следственных связей;
- предположение об ошибке.



# Тестирование «черного ящика»

**Эквивалентное разбиение** состоит в разбиении входной области данных программы на конечное число классов эквивалентности так, чтобы каждый тест, являющийся представителем некоторого класса, был эквивалентен любому другому тесту этого класса. Классы эквивалентности выделяются путем перебора входных условий и разбиения их на две (или более) группы. При этом различают два типа классов эквивалентности: правильные, задающие входные данные для программы, и неправильные, основанные на задании ошибочных входных значений.

# Тестирование «черного ящика»

Разработка тестов методом эквивалентного разбиения осуществляется в два этапа:

- *выделение классов эквивалентности*
- *построение тестов.*

При построении тестов, основанных на выборе входных данных, проводится символическое выполнение программы. Одна из целей, которая стоит перед разработчиком тестов — сокращение количества тестов, чтобы уложиться в адекватные сроки тестирования, но при этом не пропустить серьезных ошибок. Именно для этих целей используется техника разбиения на классы эквивалентности, которая и заключается в разбиении всего набора тестов на классы эквивалентности с последующим сокращением числа тестов.

# Тестирование «черного ящика»

**Анализ граничных значений** — это проверка ошибок на границах классов эквивалентности. Если техника анализа классов эквивалентности ориентирована на тестовое покрытие, то эта техника основана на рисках. Она основывается на идее о том, что программа может «дать сбой» в области граничных значений, поскольку при разработке большое число проблем возникает именно на границах входных переменных. Даже если эквивалентные классы найдены правильно, то граничные значения могут быть ошибочно отнесены к другому классу. Эффективное применение этой техники зависит от способности правильно выделить классы эквивалентности и затем выбрать тесты для проверки границ этих классов.

# Тестирование «черного ящика»

**Анализ причинно-следственных связей** происходит следующим образом. Спецификация разбивается на рабочие участки. В спецификации определяется множество причин и следствий (под причиной понимается отдельное входное условие или класс эквивалентности, следствие представляет собой выходное условие или преобразование системы). На основе анализа семантического содержания спецификации строится таблица истинности, в которой последовательно перебираются всевозможные комбинации причин и определяются следствия для каждой комбинации причин. Недостатком этого подхода является плохое исследование граничных условий.

# Тестирование «черного ящика»

**Предположение об ошибке** в значительной степени основано на интуиции. Основная идея метода состоит в том, чтобы составить список, который перечисляет возможные ошибки и ситуации, в которых эти ошибки могли проявиться. После этого на основе списка составляются тесты.

Во время подготовки к тестированию методом «черного ящика» строятся таблицы условий, причинно-следственные графы и области разбивки. Кроме того, подготавливаются тестовые наборы, учитывающие параметры и условия среды, которые влияют на поведение функций.

# Тестирование «белого ящика»

Метод «белого ящика» позволяет исследовать внутреннюю структуру программы. Тестирование по принципу «белого ящика» ориентировано на проверку прохождения всех путей программного кода.

Тестирование «белого ящика» применяется на уровне модулей путем выбора тестовых ситуаций и подготовки данных. Метод «белого ящика» включает тестирование следующих элементов:

- операторов, которые должны быть выполнены хотя бы один раз;
- путей по заданному графу потоков управления для выявления разных маршрутов передачи управления, для этого создается набор тестовых данных, гарантирующих прохождение всех путей. Однако все пути протестировать бывает невозможно, поэтому остаются не выявленные ошибки, которые могут проявиться в процессе эксплуатации;
- блоков, разделяющих программы на отдельные части, которые выполняются один раз или многократно.

# Тестирование «белого ящика»

***Тестирование «белого ящика»*** — это технология тестирования, которая чаще всего применяется на этапе разработки (иногда ее еще называют тестированием «стеклянного ящика»). Тестировщик разрабатывает тесты, основываясь на знании исходного кода, к которому он имеет полный доступ. В результате он получает следующие преимущества:

- направленность тестирования. Программист может тестировать программу по частям, разрабатывать специальные тестовые сценарии, которые вызывают тестируемый модуль и передают ему интересующие данные;
- полный охват кода. Программист всегда может определить, какие именно фрагменты кода работают в каждом тесте;

# Тестирование «белого ящика»

- возможность управления потоком команд.

Программист может включить в программу отладочные команды, отображающие информацию о ходе ее выполнения, или воспользоваться для этого специальным программным средством — отладчиком;

- возможность отслеживания целостности данных. Отслеживая состояние данных (с помощью отладчика), программист может выявить такие ошибки, как изменение данных не теми модулями, их неверная интерпретация или неудачная организация и т.п.;

- видение внутренних граничных точек. Например, для выполнения определенного действия может быть использовано несколько различных алгоритмов, но имея доступ к программному коду



# Тестирование «белого ящика»

Тестирование «белого ящика» — часть процесса программирования. Программисты выполняют эту работу постоянно, они тестируют каждый модуль после его написания, а затем еще раз — после интеграции его в систему.

Хотя тестирование «белого ящика» имеет гораздо более мощную теоретическую основу, большинство тестировщиков предпочитают тестирование «черного ящика». Тестирование «белого ящика» лучше поддается

математическому моделированию, но это совсем не означает, что оно эффективнее.

Каждая из технологий позволяет выявить ошибки, пропускаемые в случае использования другой. С этой точки зрения их можно назвать одинаково эффективными.

# Вопросы для самоконтроля:

1. Какие методы тестирования вам известны?
2. Опишите особенности методов черного, белого и серого ящиков.
3. Охарактеризуйте метод тестирования «черного ящика».
4. В чем заключается основная задача тестировщика для метода тестирования «черного ящика»?
5. Какие проблемы системы могут быть выявлены в результате тестирования методом «черного ящика»?
6. Как должны быть решены эти проблемы?
7. Какие подходы обеспечивают методы «черного ящика»?
8. В чем заключается эквивалентное разбиение?
9. Как происходит разработка тестов с помощью эквивалентного разбиения?
10. Для каких целей для используется техника разбиения на классы эквивалентности?
11. Что представляет собой анализ граничных значений?
12. В чем заключается эффективность применения техники анализа граничных значений?
13. Как происходит анализ причинно-следственных связей?
14. В чем заключается основная идея подхода «Предположение об ошибке»?
15. Охарактеризуйте метод тестирования «белого ящика».
16. Какие преимущества получает тестировщик при тестировании системы методом «белого ящика»?
17. В чем заключается эффективность тестирования системы методом «белого ящика»?
18. Какой метод тестирования предпочтительнее?