**Критерии выбора тестов**

1. **Критерий должен быть достаточным**, т.е. показывать, когда некоторое конечное множество тестов достаточно для тестирования данной программы.
2. **Критерий должен быть полным**, т.е. в случае ошибки должен существовать тест из множества тестов, удовлетворяющих критерию, который раскрывает ошибку.
3. **Критерий должен быть надежным**, т.е. любые два множества тестов, удовлетворяющих ему, одновременно должны раскрывать или не раскрывать ошибки программы
4. **Критерий должен быть легко проверяемым**, например вычисляемым на тестах

Для нетривиальных классов программ в общем случае **не существует полного и надежного критерия**, зависящего от программ или спецификаций.

Поэтому мы стремимся к идеальному общему критерию через реальные частные.

**Классы критериев**

1. **Структурные критерии-** используют информацию о структуре программы (критерии так называемого "белого ящика")

Условие критерия **тестирования команд** (критерий С0)- набор тестов в совокупности должен обеспечить прохождение каждой команды не менее одного раза. Если есть цикл- то ограничивается константой.

Условие критерия **тестирования ветвей** (критерий С1) - набор тестов в совокупности должен обеспечить прохождение каждой ветви не менее одного раза. Если есть цикл- то ограничивается константой.

Условие критерия **тестирования путей** (критерий С2) - набор тестов в совокупности должен обеспечить прохождение каждого пути не менее 1 раза. Если есть цикл- то ограничивается константой.

1. **Функциональный критерий** - Он обеспечивает, прежде всего, контроль степени выполнения требований заказчика в программном продукте. При функциональном тестировании преимущественно используется модель "черного ящика". как правило, достаточно объемны, тем не менее, соответствующая проверка должна быть всеобъемлющей.  Проблема функционального тестирования - это, прежде всего, трудоемкость; дело в том, что документы, фиксирующие требования к программному изделию.

**Тестирование пунктов спецификации** - набор тестов в совокупности должен обеспечить проверку каждого тестируемого пункта не менее одного раза. Спецификация требований может содержать сотни и тысячи пунктов требований к программному продукту и каждое из этих требований при тестировании должно быть проверено в соответствии с критерием не менее чем одним тестом

**Тестирование классов входных данных** - набор тестов в совокупности должен обеспечить проверку представителя каждого класса входных данных не менее одного раза.

**Тестирование правил** - набор тестов в совокупности должен обеспечить проверку каждого правила, если входные и выходные значения описываются набором правил некоторой грамматики.

**Тестирование классов выходных данных** - набор тестов в совокупности должен обеспечить проверку представителя каждого выходного класса, при условии, что выходные результаты заранее расклассифицированы, причем отдельные классы результатов учитывают, в том числе, ограничения на ресурсы или на время (time out).

**Тестирование функций** - набор тестов в совокупности должен обеспечить проверку каждого действия, реализуемого тестируемым модулем, не менее одного раза.

**Комбинированные критерии для программ и спецификаций** - набор тестов в совокупности должен обеспечить проверку всех комбинаций непротиворечивых условий программ и спецификаций не менее одного раза.

**3) Стохастические критерии**

Стохастическое тестирование применяется при тестировании сложных программных комплексов - когда набор детерминированных тестов (X,Y) имеет громадную мощность. В случаях, когда подобный набор невозможно разработать и исполнить на фазе тестирования, можно применить следующую методику.

* Разработать программы - имитаторы случайных последовательностей входных сигналов {x}.
* Вычислить независимым способом значения {y} для соответствующих входных сигналов {x} и получить тестовый набор (X,Y).
* Протестировать приложение на тестовом наборе (X,Y), используя два способа контроля результатов:
  + Детерминированный контроль - проверка соответствия вычисленного значения yвhttps://studfile.net/html/2706/830/html_IEb6kGd3xZ.ccRF/img-RP0psv.png{y} значению y, полученному в результате прогона теста на наборе {x} - случайной последовательности входных сигналов, сгенерированной имитатором.
  + Стохастический контроль - проверка соответствия множества значений {yв}, полученного в результате прогона тестов на наборе входных значений {x}, заранее известному распределению результатов F(Y)
* **Cтатистические методы** окончания тестирования - стохастические методы принятия решений о совпадении гипотез о распределении случайных величин. К ним принадлежат широко известные: метод Стьюдента (St), метод Хи-квадрат (χ2) и т.п.
* Метод **оценки скорости выявления ошибок** - основан на модели скорости выявления ошибок, согласно которой тестирование прекращается, если оцененный интервал времени между текущей ошибкой и следующей слишком велик для фазы тестирования приложения.
* **Метод мутационного тестирования** - в разрабатываемую программу P вносят мутации, т.е. искусственно создают программы-мутанты P1, P2... Затем программа P и ее мутанты тестируются на одном и том же наборе тестов (X,Y).

Если на наборе (X,Y) подтверждается правильность программы P и, кроме того, выявляются все внесенные в программы-мутанты ошибки, то **набор тестов (X,Y) соответствует**мутационному критерию, а тестируемая программа объявляется **правильной**.