Критерии выбора тестов. Требования к идеальному критерию тестирования и классы частных примеров

Критерий выбора тестов – решающее правило выбора

состава и содержания проверок при выполнении тестирования программного обеспечения.

Для различных классов программного обеспечения в общем случае не существует полного и надежного критерия выбора совокупности тестов, очевидно следующего из спецификаций назначения тестируемого программного обеспечения.

Целесообразно выполнять приближение к рациональному общему ("идеальному") критерию через совокупность фактических частных критериев.

Тестирование программного обеспечения

Требования к "идеальному" критерию:

-критерий должен быть достаточным, т.е. показывать, когда некоторое конечное множество тестов достаточно для тестирования данной программы;

-критерий должен быть полным, т.е. в случае ошибки должен существовать тест из множества тестов, удовлетворяющих критерию, который раскрывает ошибку;

-критерий должен быть надежным, т.е. любые два множества тестов, удовлетворяющих ему, одновременно должны раскрывать или не раскрывать ошибки программы;

-критерий должен быть легко проверяемым, например, вычисляемым на тестах.

Тестирование программного обеспечения

Классы критериев:

-структурные критерии используют информацию

оструктуре программы (критерии так называемого "белого ящика");

-функциональные критерии формулируются

в описании требований к программному изделию (критерии так называемого "черного ящика");

-критерии стохастического тестирования

формулируются в терминах проверки наличия заданных свойств у тестируемого приложения, средствами проверки некоторой статистической гипотезы;

-мутационные критерии ориентированы на проверку

свойств программного изделия на основе подхода

Монте-Карло.

–

Тестирование программного обеспечения

Структурные критерии основываются на основных элементах управляющего графа программы, операторах, ветвях, путях:

- критерий тестирования команд (критерий C0) - набор тестов в совокупности должен обеспечить прохождение каждой команды

не менее одного раза. Это слабый критерий и, как правило, используется в больших программных системах, где другие критерии применить невозможно.

- критерий тестирования ветвей (критерий C1) - набор тестов в совокупности должен обеспечить прохождение каждой ветви

не менее одного раза. Этот достаточно сильный и, при этом, экономичный критерий, поскольку множество ветвей в тестируемом приложении конечно и не велико, используется в системах автоматизации тестирования.

- критерий тестирования путей (критерий С2) - набор тестов в совокупности должен обеспечить прохождение каждого пути

не менее одного раза. Если программа содержит цикл (особенно,

с неявно заданным числом итераций), то число итераций ограничивается константой (часто - двум или количеством классов выходных путей).

Тестирование программного обеспечения

1 void Method (int *x)

{

2if (*x>17)

3 *x = 17-*x;

4if (*x==-13)

5 *x = 0;

6 }

Тестовый набор из одного теста команд (С0)

{(хвх=30, хвых=0)} покрывает все операторы трассы 1-2-3-4-5-6

Тестовый набор из двух тестов ветвей (С1)

{(30,0), (17,17)} добавляет один тест к множеству тестов для С0 и трассу 1-2-4-6. Трасса 1-2-3-4-5-6 проходит через все ветви достижимые в операторах if при условии true, а трасса 1-2-4-6 через все ветви, достижимые в операторах if при условии false.

Тестовый набор из четырех тестов путей (С2)

{ (30,0), (17,17), (-13,0), (21,-4) }

2	**if (x>17)**	>	≤	≤	>
4	**if (x==-13)**	≠	≠	=	≠
		1-2-3-4-5-6	**1-2-4-6**	**1-2-4-5-6**	**1-2-3-4-6**

Тестирование программного обеспечения

Критерий путей С2 проверяет программу более тщательно, чем критерии С1, однако даже если он удовлетворен, нет оснований утверждать, что программа реализована в соответствии со спецификацией. Например, если спецификация задает условие, что |x|<=100, невыполнимость которого можно подтвердить на тесте (-177,-177). Действительно, операторы 3 и 4 на тесте (-177,-177) не изменят величину x=-177 и результат не будет соответствовать спецификации.

Структурные критерии не проверяют соответствие спецификации, если оно не отражено в структуре программы. Поэтому при успешном тестировании программы по критерию С2 можно не заметить ошибку, связанную с невыполнением некоторых условий спецификации требований.

–

Тестирование программного обеспечения

Функциональные критерии обеспечивают контроль степени выполнения требований заказчика в программном продукте. Поскольку требования

формулируются к продукту в целом, они отражают взаимодействие тестируемого приложения с окружением.

При функциональном тестировании преимущественно используется метод "черного ящика".

Основная трудность функционального тестирования – трудоемкость работы тестировщиков, поскольку документы, фиксирующие требования к программному изделию (Software requirement specification, Functional specification и т.п.), как правило, достаточно объемны.

Тестирование программного обеспечения

Виды функциональных критериев**:**

-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.

При создании тестов классы входных данных сопоставляются

срежимами использования тестируемого компонента или подсистемы приложения, что заметно сокращает варианты перебора, учитываемые при разработке тестовых наборов.

Тестирование программного обеспечения

Виды функциональных критериев**:**

-тестирование правил - набор тестов в совокупности должен обеспечить проверку каждого правила, если входные и выходные значения описываются набором правил некоторой грамматики. Грамматика должна быть достаточно простой, чтобы трудоемкость разработки соответствующего набора тестов соответствовала срокам и штату специалистов, выделенных для реализации фазы тестирования

-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. При создании тестов классы выходных данных сопоставляются с режимами использования тестируемого компонента или подсистемы, что сокращает варианты перебора, учитываемые при разработке тестовых наборов